Journal of Machine Learning Research 22 (2021) 1-45 Submitted 5/19; Revised 1/21; Published 5/21

Collusion Detection and Ground Truth Inference in Crowdsourcing for Labeling Tasks

Changyue Song CSONG14@STEVENS.EDU
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ 07030, USA

Kaibo Liu KLIU8@WISC.EDU
Department of Industrial and Systems Engineering
University of Wisconsin-Madison
Madison, WI 53706, USA

Xi Zhang XI.ZHANG@PKU.EDU.CN
Department of Industrial Engineering and Management
Peking University
Beijing, 100871, China

Editor: Qiang Liu

Abstract

Crowdsourcing has been a prompt and cost-effective way of obtaining labels in many machine learning applications. In the literature, a number of algorithms have been developed to infer the ground truth based on the collected labels. However, most existing studies assume workers to be independent and are vulnerable to worker collusion. This paper aims at detecting the collusive behaviors of workers in labeling tasks. Specifically, we consider collusion in a pairwise manner and propose a penalized pairwise profile likelihood method based on the adaptive LASSO penalty for collusion detection. Many models that describe the behavior of independent workers can be incorporated into our proposed framework as the baseline model. We further investigate the theoretical properties of the proposed method that guarantee the asymptotic performance. An algorithm based on expectation-maximization algorithm and coordinate descent is proposed to numerically maximize the penalized pairwise profile likelihood function for parameter estimation. To the best of our knowledge, this is the first statistical model that simultaneously detects collusion, learns workers' capabilities, and infers the ground true labels. Numerical studies using synthetic and real data sets are also conducted to verify the performance of the method.

Keywords: adaptive LASSO, crowdsourcing, collusion, pairwise profile likelihood

1. Introduction

Crowdsourcing has gained its popularity as a prompt and cost-effective way of solving a variety of real-world problems, especially for those that are difficult for machines but relatively easy for human. Some applications of crowdsourcing include the reCAPTCHA system for character recognition (von Ahn et al., 2008), the online video game Foldit for protein structure prediction (Cooper et al., 2010), and Galaxy Zoo for morphological classification of galaxies (Lintott et al., 2011). Crowdsourcing also greatly benefits the machine learning

related areas (see Vaughan, 2018 for a review). For example, crowdsourcing provides an appealing way of obtaining the labels or annotations for large data sets that are otherwise difficult or expensive to obtain. These labels can be further used for supervised learning tasks such as image and video classification (Russell et al., 2008; Vondrick et al., 2013) and natural language annotation (Snow et al., 2008). In practice, crowdsourcing platforms such as Amazon Mechanical Turk (AMT) and CrowdFlower have been developed that can facilitate the collection of labeled data. For example, in AMT, the *requesters* post tasks on the platform, and then the *workers* choose to complete tasks of interest for some monetary rewards.

One major challenge of crowdsourcing is to ensure the quality of the collected labels. In particular, as the workers are often not domain experts, they are subject to mistakes. In addition, there may exist spammers, who simply provide random labels to maximize the rewards, or even adversarial workers, who intentionally provide wrong labels to mislead the result. A common strategy for improving the label quality is repeated labeling, where a task is labelled by multiple workers, and then the multiple noisy labels are aggregated to produce an estimation of the ground true label. In the literature, a number of algorithms have been developed to estimate the ground true labels and/or train the supervised learning model, see Zhang et al. (2016) for a recent review of the literature.

Most of the existing studies assume that workers do not know each other and are mutually independent; therefore, they can only consider *individual spam*. However, in practice, although some crowdsourcing platforms such as AMT do not provide means of communication, workers may still collude with each other on the same task, namely *collusive spam*, which is much more challenging to spot (Liu et al., 2017). In fact, collusion of workers is common in many crowdsourcing systems such as rating systems (Dellarocas, 2006) and question answering communities (Xu et al., 2015). For example, in 2004, Amazon's Canadian site mistakenly revealed the identities of thousands of book reviewers. It turned out that many anonymous book reviews were actually written by the author's family and friends, or from the competitors (Harmon, 2004). Gray et al. (2016) and Yin et al. (2016) pointed out that a large proportion of workers in crowdsourcing platforms such as AMT communicate and collaborate with each other via online forums to obtain more rewards. Unfortunately, most existing studies assuming independent workers are ineffective to collusive spam, which can lead to great bias in the estimated model and large errors in ground truth inference. As a simple example, consider the case of three anonymous workers with the same accuracy. If the workers are indeed independent, then majority voting is a good mechanism to infer the ground truth. However, if two workers collude with each other and give the same task label, then majority voting will always treat this label as the ground truth and thus fail to defend against the attack when the colluding workers simply give some random labels.

Despite the pressing need of a robust method for collusion detection and ground truth inference, the existing studies considering workers' collusive behaviors are still sparse and heuristic. In this study, we propose a new statistical framework for detecting the collusion of workers in crowdsourcing systems with a focus on labeling tasks, i.e., workers choose a correct answer from multiple alternatives. We call the proposed method PROCAP (Pairwise Recognition Of Collusion with Asymptotic Promise). To the best of our knowledge, this is the first statistical model that simultaneously detects the collusive behaviors of workers, learns the workers' capabilities, and infers the ground true labels. Furthermore, we prove

that under some conditions, the performance of PROCAP is theoretically guaranteed, i.e., the estimated parameters converge to the true value, and the probability for correctly detecting the collusive behaviors converges to 1. The proposed model can also be extended to other crowdsourcing tasks and systems such as rating systems.

The remainder of this paper is organized as follows. Section 2 describes the basic settings of the problem. Section 3 elaborates the model development. The theoretical properties of the proposed method are investigated in Section 4. In Section 5, we propose the algorithms for solving the optimization problem. We then discuss ground truth inference in Section 6 and review the related literature in Section 7. Numerical studies using synthetic data sets and real data sets are conducted in Section 8. At last, Section 9 draws the conclusion and discusses future work.

2. Setup

In this study, we consider a set of workers $\mathcal{W} = \{1, 2, \cdots, W\}$ and a set of tasks $\mathcal{T} = \{1, 2, \cdots, T\}$, where W and T are the total number of workers and tasks, respectively. For one task, a worker needs to choose one label from a set of alternatives $\mathcal{C} = \{1, 2, \cdots, C\}$. Denote $y_t \in \mathcal{C}$ as the unknown true label for task $t \in \mathcal{T}$, and denote $l_{i,t} \in \mathcal{C}$ as the label given by worker $i \in \mathcal{W}$ on task t. In practice, a worker may only produce labels for a subset of tasks. If worker i does not give a label for task t, we denote $l_{i,t} = 0$. Our objective is to detect the collusion of workers and infer the ground truth $\{y_t\}$ based on the collected labels $\{l_{i,t}\}$.

To detect the collusive behavior of workers, the first step is to consider the characteristics of the labels given by colluding workers. Chen et al. (2018) summarized three types of collusive behaviors:

(1) Duplicated submission, where a group of workers work together on the same task and submit the same answer;

(2) Group plagiarism, where some workers simply plagiarize others' answers; and

(3) Spam accounts, where one worker registers multiple accounts within one crowdsourcing platform and submits the same answer to a task for multiple times, which is often referred to as Sybil attack (Douceur, 2002).

In all the three types of collusive behaviors, the colluding workers will give the same label on a task, which is the key for collusion detection. However, great challenges still exist. First, to avoid being detected, colluding workers may choose to collude on only a fraction of tasks, making the collusive behavior more difficult to spot. Second, the effect of collusion can be confounded with other factors as independent workers can also give the same label on a task. For example, if two workers have high expertise and always give the right answer, their labels will always be the same. Thus, it is very important to distinguish duplicated labels resulted from collusive behavior and from other factors. Third, a worker may collude with different workers on different tasks, leading to a large number of possible colluding scenarios. To address the challenges, we propose to formulate the problem in an innovative pairwise manner, as described in the next section.

3. Collusion Detection

To model worker collusion, it is important to distinguish the labels generated from independent workers and those from colluding workers. To address this issue, we simultaneously model the behaviors of independent workers and colluding workers. At first, we select a baseline model which characterizes the behavior of independent workers.

3.1 The Baseline Models

Without loss of generality, as a demonstration, we consider the General Dawid-Skene (DS) model as our baseline model (Dawid and Skene, 1979). In fact, the General DS model is very flexible and by imposing some restrictions, it reduces to various models that are commonly used in the literature. Here we give a brief introduction to these models (Li and Yu, 2014).

- *General DS model.* This model assumes when worker i works independently, for a task with true label Y, the label L_i is generated according to a C-by-C confusion matrix $\boldsymbol{a}_i$, where the entry at the yth row and lth column is

$$[\boldsymbol{a}_i]_{y,l} = p(L_i = l | Y = y) \ ,$$

  and $\sum_{l \in \mathcal{C}} [\boldsymbol{a}_i]_{y,l} = 1$. As can be seen, $\boldsymbol{a}_i$ is task-independent, which means an independent worker is assumed to adopt the same confusion matrix for all tasks. Here we drop the subscript t for simplicity and use the upper letters L_i, Y to represent the corresponding random variables for a general task.

- *Class-Dependent DS model.* For each row of the confusion matrix $\boldsymbol{a}_i$ in the General DS model, if we restrict the off-diagonal entries to be the same, i.e.,

$$[\boldsymbol{a}_i]_{y,l} = \frac{1 - [\boldsymbol{a}_i]_{y,y}}{C - 1} \ , \text{if } l \neq y \ ,$$

  then the General DS model reduces to the Class-Dependent DS model. A special case is that for binary labeling tasks with $C = 2$, the Class-Dependent DS model is the same as the General DS model, which is also referred to as the *two-coin model*.

- *Homogeneous DS model.* For each confusion matrix $\boldsymbol{a}_i$ in the Class-Dependent DS model, if we further restrict the diagonal entries to be the same, i.e.,

$$[\boldsymbol{a}_i]_{y,l} = \begin{cases} a_i, & \text{if } y = l \ , \\ \frac{1-a_i}{C-1}, & \text{if } y \neq l \ , \end{cases}$$

  then Class-Dependent DS model reduces to the Homogeneous DS model. For binary labeling tasks with $C = 2$, the Homogeneous DS model is also referred to as the *one-coin model*.

- *Uniform DS model.* Following the same spirit, we can further simplify the Homogeneous DS model by restricting the confusion matrix of each worker to be the same, i.e.,

$$[\boldsymbol{a}_i]_{y,l} = \begin{cases} a, & \text{if } y = l \ , \\ \frac{1-a}{C-1}, & \text{if } y \neq l \ . \end{cases}$$

We call it the Uniform DS model throughout this paper. In fact, there is a connection between the Uniform DS model and majority voting, which will be elaborated in Section 8.

Besides the models mentioned above, other existing models such as the GLAD model (Generative model of Labels, Abilities, and Difficulties) proposed by Whitehill et al. (2009) can also be used as the baseline. In this section, we consider the General DS model for demonstration due to its flexibility, and other models can be incorporated in a similar way. The procedure for incorporating the GLAD model is briefly introduced in Appendix A as another example. Next, we discuss a novel approach to model worker collusion based on the pairwise likelihood.

3.2 Pairwise Likelihood Estimation

In the literature, collusion detection is usually formulated as a problem of partitioning the set of workers $\mathcal{W}$ into several mutually exclusive groups, where the workers in the same group collude with each other (Liu et al., 2008; Xu, 2013). However, this formulation is not flexible enough to realize the practical issue when a worker colludes with different workers on different tasks. In addition, since the partition is a discrete formulation, deriving the optimal partition of workers usually requires much computation. To address the challenges, we consider the workers in a pairwise manner. Specifically, we propose a Bernoulli random variable $z^t_{i,j} \sim \text{Bernoulli}(H_{i,j})$ representing whether worker i and worker j collude on task t, where $z^t_{i,j} = 1$ means worker i and worker j collude on task t, and $z^t_{i,j} = 0$ otherwise. $H_{i,j}$ represents the probability that worker i and worker j collude. According to the definition, the variables $z^t_{i,j}$ and $H_{i,j}$ are symmetric regarding to i and j in that $z^t_{i,j} = z^t_{j,i}$ and $H_{i,j} = H_{j,i}$. In this way, the formulation is very flexible to represent various colluding scenarios of workers. For example, to represent the scenario when workers 1, 2, and 3 collude together on task t, we have $z^t_{1,2} = z^t_{1,3} = z^t_{2,3} = 1$. For another example, if worker 1 colludes with worker 2 on task 1, and colludes with worker 3 on task 2, we can represent this case as $z^1_{1,2} = 1, z^2_{1,3} = 1$. In addition, we can capture the situation when workers only collude on a portion of tasks using a value of $H_{i,j} < 1$. Note an implicit constraint is that $z^t_{i,j}$ is transitive. In other words, for a certain task t, the matrix $[z^t_{i,j}]_{i,j\in\mathcal{W}}$ can be permuted as block diagonal by rearranging the sequence of worker index i, j. For example, if $z^t_{1,2} = z^t_{2,3} = 1$, then we must have $z^t_{1,3} = 1$. This is because if worker 1 colludes with worker 2 and worker 2 colludes with worker 3 on task t, then worker 1 cannot be independent with worker 3, even if worker 1 does not directly collude with worker 3 on this task. Nevertheless, as will be discussed later, the implicit constraint on transitive $z^t_{i,j}$ can be relaxed for parameter estimation.

In our proposed model, we simultaneously consider the behaviors of colluding workers and independent workers. In this way, when some workers generate the same label, our method is able to distinguish whether they are independent or colluding by deciding which situation is more probable. Specifically, we make two assumptions depending on whether two workers collude or not. At first, the *colluding assumption* is that if worker i and worker j collude on a task, they always generate the same label according to Chen et al. (2018), i.e.,

$$p(L_i = l, L_j = l' | Z_{i,j} = 1, Y = y) = I(l = l') p(L_i = L_j = l | Z_{i,j} = 1, Y = y) \ .$$

Here we again drop the subscript t for simplicity and use the upper letters $L_i, L_j, Z_{i,j}, Y$ to represent the corresponding random variables for a general task. The indicator function $I(l = l') = 1$ if $l = l'$, and otherwise $I(l = l') = 0$. To model the probability $p(L_i = L_j = l|Z_{i,j} = 1, Y = y)$, one possible way is to assume a new confusion matrix $\boldsymbol{b}_{i,j}$ for the pair of workers (i, j), where the entry at the yth row and lth column is $[\boldsymbol{b}_{i,j}]_{y,l} = p(L_i = L_j = l|Z_{i,j} = 1, Y = y)$. The second assumption, the *non-colluding assumption* is that if worker i and worker j do not collude on a task, they generate labels according to their own confusion matrices, i.e.,

$$p(L_i = l, L_j = l'|Z_{i,j} = 0, Y = y) = p(L_i = l|Y = y)p(L_j = l'|Y = y) \ . \tag{1}$$

It is worth noting that in practice, the *non-colluding assumption* might be an approximation. For example, even if worker i does not collude with worker j on a task, he/she may collude with another worker k and generate the label according to a confusion matrix $\boldsymbol{b}_{i,k}$, where $\boldsymbol{b}_{i,k} \neq \boldsymbol{a}_i$. There are two general cases when the *non-colluding assumption* is a good approximation. In the first case, each worker generates labels according to a similar confusion matrix no matter whether the worker is working independently or colluding with others. Specifically, in the previous example, the first case means that when $\boldsymbol{b}_{i,k}$ is close to $\boldsymbol{a}_i$, (1) will hold for worker i and worker j. In the second case, the colluding workers collude frequently and the confusion matrix when they collude can be also regarded as their own confusion matrices. Specifically, if worker i colludes frequently with worker k with a probability $H_{i,k}$ close to 1, then only the value of $\boldsymbol{b}_{i,k}$ matters, which means that we can simply regard $\boldsymbol{a}_i$ and $\boldsymbol{a}_k$ to be the same as $\boldsymbol{b}_{i,k}$, and then (1) will hold for worker i and worker j.

Under the two assumptions, we formulate the problem using pairwise likelihood approach (Varin et al., 2011). Denote $\mathcal{P} = \{(i, j) : i, j \in \mathcal{W}, i < j\}$ as the set of worker pairs. In practice, it is possible that two workers generate labels for mutually exclusive tasks only, i.e., $\nexists t$ such that $l_{i,t} \neq 0$ and $l_{j,t} \neq 0$. In this case, we exclude this pair from $\mathcal{P}$. Let $\tilde{\boldsymbol{\theta}}_{i,j} = (\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{b}_{i,j}, \boldsymbol{m}, H_{i,j})$ be the model parameters involved for the pair of worker i and worker j, where $\boldsymbol{m} = [m_1, \cdots, m_C]^T$ is the vector of marginal probabilities of y_t with $m_c = p(y_t = c)$, and let $\tilde{\boldsymbol{\theta}} = \cup_{(i,j)\in\mathcal{P}}\tilde{\boldsymbol{\theta}}_{i,j}$ be the set of all model parameters. Specifically, given the labels $\{l_{i,t}, i \in \mathcal{W}, t \in \mathcal{T}\}$, the log-likelihood for the pair of workers (i, j) is

$$\ell_p^{(i,j)}(\tilde{\boldsymbol{\theta}}_{i,j}) = \sum_t \log p(L_i = l_{i,t}, L_j = l_{j,t}|\tilde{\boldsymbol{\theta}}_{i,j}) \ .$$

Then the overall pairwise log-likelihood is

$$\ell_p(\tilde{\boldsymbol{\theta}}) = \sum_{(i,j)\in\mathcal{P}} \sum_t \log p(L_i = l_{i,t}, L_j = l_{j,t}|\tilde{\boldsymbol{\theta}}_{i,j}) \ . \tag{2}$$

Since $l_{i,t}, l_{j,t} \in \mathcal{C}$, we can define $n_{i,j}^{l,l'} = \sum_{t\in\mathcal{T}} I(l_{i,t} = l) \cdot I(l_{j,t} = l')$ as the number of tasks where workers i and j generate a label of l and l', respectively. Then (2) can be rewritten as

$$\ell_p(\tilde{\boldsymbol{\theta}}) = \sum_{(i,j)\in\mathcal{P}} \sum_{l,l'\in\mathcal{C}} n_{i,j}^{l,l'} \log p(L_i = l, L_j = l'|\tilde{\boldsymbol{\theta}}_{i,j}) \ ,$$

where

$$p(L_i = l, L_j = l' | \tilde{\boldsymbol{\theta}}_{i,j}) = \sum_{y \in \mathcal{C}} [\boldsymbol{a}_i]_{y,l} [\boldsymbol{a}_j]_{y,l'} m_y (1 - H_{i,j}) + \sum_{y \in \mathcal{C}} I(l = l') [\boldsymbol{b}_{i,j}]_{y,l} m_y H_{i,j} \ .$$

To estimate the parameters, we can maximize the pairwise likelihood function $\ell_p(\tilde{\boldsymbol{\theta}})$ such that all parameters of $\tilde{\boldsymbol{\theta}}$ are between 0 and 1, $\sum_{y \in \mathcal{C}} m_y = 1$, $\sum_{l \in \mathcal{C}} [\boldsymbol{a}_i]_{y,l} = 1$, and $\sum_{l \in \mathcal{C}} [\boldsymbol{b}_{i,j}]_{y,l} = 1$. A benefit of using the pairwise likelihood approach is that the implicit constraint on transitive $z^t_{i,j}$ can be relaxed. This is because the constraint involves three or more workers and does not apply to a pair of workers. Appendix C.1 provides further discussions.

However, there are critical issues with the pairwise likelihood estimation method. First, the model is over-parameterized by assigning a confusion matrix to each pair of workers. An intuitive way to address this issue is to impose constraints on $\boldsymbol{b}_{i,j}$, for example, by assuming $\boldsymbol{b}_{i,j} = \frac{1}{2}(\boldsymbol{a}_i + \boldsymbol{a}_j)$. However, this approach restricts the model flexibility with more assumptions. Second, the identifiability of $\boldsymbol{b}_{i,j}$ depends on the value of $H_{i,j}$. Specifically, $\boldsymbol{b}_{i,j}$ is only identifiable when $H_{i,j} > 0$, which may cause numerical issues in solving the optimization problem. To address these issues, in the next subsection, we propose a penalized pairwise profile likelihood method for collusion detection.

3.3 Penalized Pairwise Profile Likelihood Estimation

To address the theoretical and numerical issues caused by the parameters $\boldsymbol{b}_{i,j}$ in pairwise likelihood estimation, our innovated idea is to adopt the profile likelihood and eliminate the parameters $\boldsymbol{b}_{i,j}$ from the objective function. Specifically, we find the upper bound of the log-likelihood function $\ell_p(\tilde{\boldsymbol{\theta}})$, denoted as $\bar{\ell}_p(\boldsymbol{\theta})$, such that $\bar{\ell}_p(\boldsymbol{\theta}) \geq \ell_p(\tilde{\boldsymbol{\theta}})$, where $\boldsymbol{\theta} = \cup_{(i,j) \in \mathcal{P}} \boldsymbol{\theta}_{i,j}$, and $\boldsymbol{\theta}_{i,j} = (\boldsymbol{a}_i, \boldsymbol{a}_j, \boldsymbol{m}, H_{i,j})$ denotes the parameters without $\boldsymbol{b}_{i,j}$. Consequently, we propose to maximize the profile log-likelihood

$$\bar{\ell}_p(\boldsymbol{\theta}) = \sum_{(i,j) \in \mathcal{P}} \bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) \ ,$$

where

$$\bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) = \sum_{l,l' \in \mathcal{C}: l \neq l'} n_{i,j}^{l,l'} \log p(L_i = l, L_j = l', Z_{i,j} = 0 | \boldsymbol{\theta}_{i,j}) + \left(\sum_{l \in \mathcal{C}} n_{i,j}^{l,l} \right) \log p(L_i = L_j | \boldsymbol{\theta}_{i,j}) \ , \quad (3)$$

such that all parameters of $\boldsymbol{\theta}$ are between 0 and 1, $\sum_{y \in \mathcal{C}} m_y = 1$, and $\sum_{l \in \mathcal{C}} [\boldsymbol{a}_i]_{y,l} = 1$. Here

$$p(L_i = l, L_j = l', Z_{i,j} = 0 | \boldsymbol{\theta}_{i,j}) = (1 - H_{i,j}) \sum_{y \in \mathcal{C}} [\boldsymbol{a}_i]_{y,l} [\boldsymbol{a}_j]_{y,l'} m_y \ ,$$

$$p(L_i = L_j | \boldsymbol{\theta}_{i,j}) = H_{i,j} + (1 - H_{i,j}) \sum_{y,l \in \mathcal{C}} [\boldsymbol{a}_i]_{y,l} [\boldsymbol{a}_j]_{y,l} m_y \ .$$

The details for deriving the pairwise profile log-likelihood function can be found in Appendix B. The rationale is that, for each pair of workers, we distinguish the tasks that receive the

same label from those receiving different labels. According to the *colluding assumption*, if worker i and worker j generate different labels on a task, they must be independent of each other on the task, which means $\boldsymbol{b}_{i,j}$ is not related to the probability $p(L_i \neq L_j|\boldsymbol{\theta}_{i,j})$. For the case when workers i and j generate the same label, we only consider the overall probability $p(L_i = L_j|\boldsymbol{\theta}_{i,j})$ instead of the probabilities $p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j}), l \in \mathcal{C}$. Since $p(L_i = L_j|\boldsymbol{\theta}_{i,j}) = 1 - p(L_i \neq L_j|\boldsymbol{\theta}_{i,j})$ is also independent of $\boldsymbol{b}_{i,j}$, the resulting pairwise profile log-likelihood does not involve $\boldsymbol{b}_{i,j}$.

As another point of view, $\bar{\ell}_p(\boldsymbol{\theta})$ can also be regarded as an M-estimator

$$\bar{\ell}_p(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{P}} \sum_{t\in\mathcal{T}} \mathcal{M}_{i,j}(l_{i,t}, l_{j,t}; \boldsymbol{\theta}_{i,j}) \ , \tag{4}$$

where

$$\mathcal{M}_{i,j}(l_{i,t}, l_{j,t}; \boldsymbol{\theta}_{i,j}) = \begin{cases} \log p(L_i = l_{i,t}, L_j = l_{j,t}, Z_{i,j} = 0|\boldsymbol{\theta}_{i,j}), & \text{if } l_{i,t} \neq l_{j,t} \ , \\ \log\left[H_{i,j} + (1 - H_{i,j})p(L_i = L_j|Z_{i,j} = 0, \boldsymbol{\theta}_{i,j})\right], & \text{if } l_{i,t} = l_{j,t} \ . \end{cases} \tag{5}$$

In Appendix C.1, we show that the true parameters $\boldsymbol{\theta}^*_{i,j}$ is a maximizer of $E_{\boldsymbol{\theta}^*_{i,j}}[\mathcal{M}_{i,j}(l_{i,t}, l_{j,t}; \boldsymbol{\theta}_{i,j})]$, which means that we can estimate the parameters by maximizing $\bar{\ell}_p(\boldsymbol{\theta})$. From the above equation, it is straightforward to see that we discard some information. Specifically, we only consider the overall probability for two workers to generate the same label, but neglect the exact value of the label. In other words, we do not distinguish the cases $l_{i,t} = l_{j,t} = l$ and $l_{i,t} = l_{j,t} = l'$ for any $l, l' \in \mathcal{C}$. In this way, we eliminate the parameters $\boldsymbol{b}_{i,j}$ at a cost of discarding this information. In practice, discarding the information may cause additional identifiability issues of the model parameters, but this can be alleviated by blending the likelihood and the profile likelihood, as will be discussed in Section 5.2.

In practice, it is probable that many workers do not collude with each other, which means that $H_{i,j}$ should be sparse. To keep the model parsimonious and avoid false detection of collusion, we apply a penalty function to the objective function and maximize

$$\bar{f}(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{P}} \bar{f}^{(i,j)}(\boldsymbol{\theta}_{i,j}) = \bar{\ell}_p(\boldsymbol{\theta}) - \sum_{(i,j)\in\mathcal{P}} n_{i,j}\mathcal{J}_\lambda(H_{i,j}) \ ,$$

where

$$\bar{f}^{(i,j)}(\boldsymbol{\theta}_{i,j}) = \bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) - n_{i,j}\mathcal{J}_\lambda(H_{i,j}) \ .$$

Here $\lambda > 0$ is the tuning parameter, $n_{i,j} = \sum_{l,l'\in\mathcal{C}} n_{i,j}^{l,l'}$ is the total number of the common tasks that both worker i and worker j finish, and $\mathcal{J}_\lambda(\cdot)$ is the penalty function. In this formulation, we use $n_{i,j}$ to weight the penalty. In this way, pairs of workers with larger $n_{i,j}$ receive larger penalties.

Commonly used penalty functions such as the Least Absolute Shrinkage and Selection Operator (LASSO) and the Smoothly Clipped Absolute Deviation (SCAD) penalty can be adopted here (Tibshirani, 1996; Fan and Li, 2001). In this study, we focus on the adaptive LASSO penalty because it shows appealing theoretical properties (Zou, 2006) and can achieve superior performance for variable selection in different applications (Zou et al., 2011; Kamarianakis et al., 2012; Kim et al., 2019; Song et al., 2019). With the adaptive

LASSO penalty $\mathcal{J}_\lambda(H_{i,j}) = \lambda w_{i,j} H_{i,j}$, where $w_{i,j}$ is a user-specified weight for the pair of workers (i,j), our objective function becomes

$$\bar{f}(\boldsymbol{\theta}) = \bar{\ell}_p(\boldsymbol{\theta}) - \lambda \sum_{(i,j)\in\mathcal{P}} n_{i,j} w_{i,j} H_{i,j} \ .$$

According to Zou (2006), $w_{i,j}$ can be estimated by $\hat{w}_{i,j} = 1/\tilde{H}^r_{i,j}$, where $\tilde{H}_{i,j}$ is the estimation of $H_{i,j}$ obtained by maximizing the pairwise profile likelihood function $\bar{\ell}_p(\boldsymbol{\theta})$ without penalty, and r is a pre-specified positive number. In this study, we use $r = 1$ for the numerical experiments. The tuning parameter λ is usually selected using cross validation or information criteria. In our case, the cross validation may not be applicable because different workers may generate labels for different tasks, and as a result, we may not be able to find a partition of tasks that can be effectively used for cross validation for all pairs of workers. Therefore, we rely on information criteria for selecting λ. Specifically, since $\bar{\ell}_p(\boldsymbol{\theta})$ can be decomposed into local models $\bar{\ell}_p(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{P}} \bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ for each pair of workers $(i,j) \in \mathcal{P}$, we can define information criteria for each local model in the conventional way and sum up to obtain the overall information criteria for $\bar{\ell}_p(\boldsymbol{\theta})$. Consequently, we define the Akaike information criterion (AIC) of $\boldsymbol{\theta}$ as

$$\mathrm{AIC}_{\boldsymbol{\theta}} = -2\bar{\ell}_p(\boldsymbol{\theta}) + 2 \sum_{(i,j)\in\mathcal{P}} I(H_{i,j} \neq 0) \ ,$$

and the Bayesian information criterion (BIC) as

$$\mathrm{BIC}_{\boldsymbol{\theta}} = -2\bar{\ell}_p(\boldsymbol{\theta}) + \sum_{(i,j)\in\mathcal{P}} I(H_{i,j} \neq 0) \cdot \log n_{i,j} \ .$$

Then, we can select a λ such that the corresponding estimated $\boldsymbol{\theta}$ has the minimum AIC or BIC.

4. Asymptotic Properties

In this section, we discuss the asymptotic properties of the proposed PROCAP method. At first, we reparametrize the model to remove the constraints $\sum_{l=1}^{C} [\boldsymbol{a}_i]_{y,l} = 1$ and $\sum_{y=1}^{C} m_y = 1$. Specifically, we replace $[\boldsymbol{a}_i]_{y,C}$ by $1 - \sum_{l=1}^{C-1} [\boldsymbol{a}_i]_{y,l}$ and replace m_C by $1 - \sum_{y=1}^{C-1} m_y$. For simplicity, we still use $\boldsymbol{\theta}$ to represent the parameters after re-parametrization. The first issue we need to consider is the identifiability of $\boldsymbol{\theta}$. Mathematically, $\boldsymbol{\theta}$ is generally not identifiable. A simple example is that for $C = 2$, if we exchange the two entries of $\boldsymbol{m}$, and exchange the two rows for each $\boldsymbol{a}_i$, then $\bar{\ell}_p(\boldsymbol{\theta})$ remains unchanged. However, in practice, external information or domain knowledge is usually available to narrow down the space of $\boldsymbol{\theta}$ such that the true parameter $\boldsymbol{\theta}^*$ is identifiable. For the previous example, if worker i is known to be honest, or the first entry of $\boldsymbol{m}^*$ is known to be greater than the second one, we can impose proper restrictions such as $[\boldsymbol{a}_i]_{1,1} > [\boldsymbol{a}_i]_{1,2}$ and $m_1 > m_2$ accordingly when estimating $\boldsymbol{\theta}$. Throughout this section, we assume that $\boldsymbol{\theta}^*$ is identifiable. In addition, for simplicity, we assume each worker generates a label for each task, i.e., $n_{i,j} = T$ for any $(i,j) \in \mathcal{P}$, where T is the total number of tasks. In fact, the asymptotic properties

discussed below apply to the local model $\bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ and $\bar{f}^{(i,j)}(\boldsymbol{\theta}_{i,j})$ for each pair of workers, which means that the proposed PROCAP method still enjoys these properties even when different pairs of workers generate labels for different number of tasks. Further, it should be noted that the asymptotic properties discussed in this section are under the regularity condition that each entry of the true $\boldsymbol{a}_i^*$ and $\boldsymbol{m}^*$ is within the interval $(0,1)$ and each true $H_{i,j}^*$ is within the interval $[0,1)$, i.e., the parameter $\boldsymbol{\theta}^*$ is allowed to be on the boundary with $H_{i,j}^* = 0$. This is different from many existing studies where the true parameter is usually assumed to be an inner point.

In this section, we present the main results and the proofs can be found in Appendix C. At first, we consider the estimated $\hat{\boldsymbol{\theta}}$ by maximizing $\bar{\ell}_p(\boldsymbol{\theta})$ without penalty. As mentioned in Section 3.3, $\bar{\ell}_p(\boldsymbol{\theta})$ can be viewed as an M-estimator. Then, we can show the following theorem.

Theorem 1 (Consistency of pairwise profile likelihood) *$\hat{\boldsymbol{\theta}} = \operatorname{argmax}_{\boldsymbol{\theta}} \bar{\ell}_p(\boldsymbol{\theta})$ is root-n consistent, i.e., $\sqrt{T}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) = O_p(1)$.*

Note that throughout the section and Appendix C, n refers to the number of tasks T. Usually, the maximum likelihood estimator is asymptotically normal. However, in our case, the true $\boldsymbol{\theta}^*$ may be on the boundary, i.e., the true $H_{i,j}^* = 0$. Thus, the estimated $\hat{\boldsymbol{\theta}}$ may not be asymptotically normal, but it is still consistent with the common root-n convergence rate.

Next, we consider the asymptotic properties of the penalized pairwise profile likelihood estimation method. Let $\mathcal{P}_0^* = \{(i,j) \in \mathcal{P} : H_{i,j}^* = 0\}$ be the set of worker pairs with the true $H_{i,j}^*$ to be 0, and let $\mathcal{P}_1^* = \mathcal{P} \backslash \mathcal{P}_0^* = \{(i,j) \in \mathcal{P} : H_{i,j}^* > 0\}$ be the complement set of $\mathcal{P}_0^*$. To emphasize that the selection of the tuning parameter λ depends on the number of tasks T, we explicitly write the tuning parameter as λ_T. In addition, denote $g_T^0 = \min\{\lambda_T w_{i,j}, (i,j) \in \mathcal{P}_0^*\}$ and $g_T^1 = \max\{\lambda_T w_{i,j}, (i,j) \in \mathcal{P}_1^*\}$ as the minimum and maximum penalty coefficients of $\lambda_T w_{i,j}$ for non-colluding and colluding worker pairs, respectively. The next theorem verifies the consistency of the penalized pairwise profile likelihood method.

Theorem 2 (Consistency of penalized pairwise profile likelihood) *If $\sqrt{T} g_T^1 \to 0$, there exists a local maximizer $\hat{\boldsymbol{\theta}}$ of $\bar{f}(\boldsymbol{\theta})$ such that $\sqrt{T}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) = O_p(1)$.*

This theorem indicates that as long as the maximum penalty coefficient for colluding workers shrinks to 0 at a rate faster than $1/\sqrt{T}$, there exists a local maximizer of the penalized pairwise profile likelihood that is root-n consistent.

In fact, the root-n consistent local maximizer possesses the oracle property. Specifically, if we divide the parameter $\boldsymbol{\theta}$ into two parts as $\boldsymbol{\theta} = (\boldsymbol{\theta}_0, \boldsymbol{\theta}_c)$, where $\boldsymbol{\theta}_0 = \{H_{i,j}, (i,j) \in \mathcal{P}_0^*\}$ and $\boldsymbol{\theta}_c = \boldsymbol{\theta} \backslash \boldsymbol{\theta}_0$, the oracle property is stated in the following theorem.

Theorem 3 (Oracle property) *If $\sqrt{T} g_T^1 \to 0$, $\sqrt{T} g_T^0 \to \infty$, then with probability tending to 1, the root-n consistent local maximizer $\hat{\boldsymbol{\theta}}$ of $\bar{f}(\boldsymbol{\theta})$ in Theorem 2 satisfies:*

(a) Selection consistency: $\boldsymbol{\theta}_0 = \mathbf{0}$; and

(b) Asymptotic normality: $\sqrt{T}(\hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c^*) = N(\mathbf{0}, \boldsymbol{\Sigma}_c)$, where $\boldsymbol{\Sigma}_c$ is a constant covariance matrix.

This theorem indicates that with probability goes to 1, the estimated $\boldsymbol{\theta}_0$ is $\mathbf{0}$ and the estimated $\hat{\boldsymbol{\theta}}_c$ is normally distributed with mean $\boldsymbol{\theta}_c^*$.

As mentioned before, we estimate $w_{i,j}$ by $\hat{w}_{i,j} = 1/\tilde{H}_{i,j}^r$, where $\tilde{H}_{i,j}$ is the estimation of $H_{i,j}$ without penalty, and r is a positive number. Since $\tilde{H}_{i,j}$ is root-n consistent according to Theorem 1, it is easy to verify that as long as $T^{1/2}\lambda_T \to 0$ and $T^{(1+r)/2}\lambda_T \to \infty$, the technical conditions for Theorem 2 and Theorem 3 are satisfied. For example, if $r = 1$ as considered in this paper, then $\lambda_T = \frac{1}{T}\log T$ satisfies the technical conditions. However, selecting λ in this way may not be optimal in practice, because it simply regards λ as a deterministic function of T and does not consider the collected data. As mentioned before, we may consider selecting λ according to information criteria.

Next, we show that with probability tending to 1, any λ that fails to identify the correct model will not be selected by BIC. Denote $\hat{\boldsymbol{\theta}}(\lambda)$ as the corresponding estimation of the penalized pairwise profile likelihood method with tuning parameter λ. Let $\hat{\mathcal{P}}_0(\lambda) = \{(i,j) \in \mathcal{P} : \hat{H}_{i,j}(\lambda) = 0\}$ be the collection of worker pairs where the two workers are independent according to $\hat{\boldsymbol{\theta}}(\lambda)$, and let $\hat{\mathcal{P}}_1(\lambda) = \mathcal{P} \backslash \hat{\mathcal{P}}_0(\lambda) = \{(i,j) \in \mathcal{P} : \hat{H}_{i,j}(\lambda) > 0\}$. We can partition the set of nonnegative numbers $\mathcal{R}_+ = [0, +\infty)$ into three mutually exclusive sets $\mathcal{R}_+ = \mathcal{R}_+^u \cup \mathcal{R}_+^c \cup \mathcal{R}_+^o$, where for any $\lambda \in \mathcal{R}_+^u$, the model is underfitted with $\hat{\mathcal{P}}_0(\lambda) \not\subset \mathcal{P}_0^*$; for any $\lambda \in \mathcal{R}_+^c$, the model is correct with $\hat{\mathcal{P}}_0(\lambda) = \mathcal{P}_0^*$; and for any $\lambda \in \mathcal{R}_+^o$, the model is overfitted with $\hat{\mathcal{P}}_0(\lambda) \subset \mathcal{P}_0^*, \hat{\mathcal{P}}_0(\lambda) \neq \mathcal{P}_0^*$. In addition, denote $\{\tilde{\lambda}_T\}$ as a sequence of λ that satisfies the technical conditions, e.g., $\tilde{\lambda}_T = \frac{1}{T}\log T$. Then the corresponding local maximizer $\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)$ satisfies the properties in Theorem 2 and 3. Using $\tilde{\lambda}_T$ as reference, we show that if λ leads to an underfitted or overfitted model, then with probability tending to 1, the BIC of $\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)$ is smaller than the BIC of $\hat{\boldsymbol{\theta}}(\lambda)$, and thus λ will not be selected by the BIC criterion.

Theorem 4 (BIC selection consistency)

$$p\left(\inf_{\lambda \in \mathcal{R}_+^u \cup \mathcal{R}_+^o} BIC_{\hat{\boldsymbol{\theta}}(\lambda)} > BIC_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)}\right) \to 1 \ .$$

This theorem indicates that asymptotically, the model selected by BIC is the correct model.

5. Numerical Algorithm

In this section, we discuss the algorithms to maximize $\bar{\ell}_p(\boldsymbol{\theta})$ and $\bar{f}(\boldsymbol{\theta})$.

5.1 Maximizing Pairwise Profile Likelihood

Expectation-maximization (EM) algorithm is usually used to maximize the likelihood function with missing data. In our study, to maximize the pairwise likelihood $\bar{\ell}_p(\boldsymbol{\theta})$, we adopt the EM algorithm for composite likelihood proposed by Gao and Song (2011). Specifically,

we define the Q-function in the expectation step as

$$\begin{aligned}
&\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k) \\
&= \sum_{(i,j)\in\mathcal{P}} \Bigg[\sum_{l,l'\in\mathcal{C}:l\neq l'} n_{i,j}^{l,l'} \sum_{y\in\mathcal{C}} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) \log p(L_i=l, L_j=l', Y=y, Z_{i,j}=0|\boldsymbol{\theta}_{i,j}) \\
&+ \left(\sum_{l\in\mathcal{C}} n_{i,j}^{l,l}\right) \sum_{y,l\in\mathcal{C}} \sum_{z\in\{0,1\}} \mathcal{Q}_{i,j}^2(y,z,l|\boldsymbol{\theta}_{i,j}^k) \log p(L_i=L_j=l, Y=y, Z_{i,j}=z|\boldsymbol{\theta}_{i,j}) \Bigg] ,
\end{aligned}$$

where

$$\mathcal{Q}_{i,j}^1(y|z,l,l',\boldsymbol{\theta}_{i,j}^k) = p(Y=y|Z_{i,j}=z, L_i=l, L_j=l', \boldsymbol{\theta}_{i,j}^k)$$

and

$$\mathcal{Q}_{i,j}^2(y,z,l|\boldsymbol{\theta}_{i,j}^k) = p(Y=y, Z_{i,j}=z, L_i=L_j=l|L_i=L_j, \boldsymbol{\theta}_{i,j}^k)$$

are constants with respect to $\boldsymbol{\theta}$, and $\boldsymbol{\theta}_{i,j}^k$ is the estimation of $\boldsymbol{\theta}$ at the kth iteration. Our definition of the Q-function is slightly different from the literature, because our objective function is the profile likelihood function instead of the likelihood function. In this Q-function, besides y_t and $z_{i,j}^t$, we also consider different possible scenarios when $L_i = L_j$ as a latent variable, i.e., we distinguish the scenarios that $L_i = L_j = l$ for different l. In this way, we can maximize $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$ by separately considering the parameters $H_{i,j}$, $\boldsymbol{a}_i$, and $\boldsymbol{m}$. In Appendix D, we prove the following theorem, which verifies that we can maximize $\bar{\ell}_p(\boldsymbol{\theta})$ by iteratively maximizing $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$.

Theorem 5 (Verification of EM algorithm) *If* $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k) \geq \mathcal{Q}(\boldsymbol{\theta}^k|\boldsymbol{\theta}^k)$, *then* $\bar{\ell}_p(\boldsymbol{\theta}) \geq \bar{\ell}_p(\boldsymbol{\theta}^k)$.

Furthermore, it is straightforward to see that $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$ contains the term

$$\begin{aligned}
&\log p(L_i=L_j=l, Y=y, Z_{i,j}=1|\boldsymbol{\theta}_{i,j}) = \\
&\qquad \log m_y + \log H_{i,j} + \log p(L_i=L_j=l|Y=y, Z_{i,j}=1) ,
\end{aligned}$$

where the term $\log p(L_i = L_j = l|Y = y, Z_{i,j} = 1)$ is unrelated to $\boldsymbol{\theta}$, because $\boldsymbol{\theta}$ does not contain any parameter that specifies the behavior of two workers when they collude. Therefore, we simply omit this term when maximizing $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$. As a result, we obtain the following updating equations for each parameter

$$H_{i,j}^{k+1} = \frac{\sum_{l\in\mathcal{C}} n_{i,j}^{l,l}}{\sum_{l,l'\in\mathcal{C}} n_{i,j}^{l,l'}} \cdot \frac{H_{i,j}^k}{H_{i,j}^k + p(L_i=L_j|Z_{i,j}=0, \boldsymbol{\theta}_{i,j}^k)(1-H_{i,j}^k)} ,$$

$$m_y^{k+1} \propto \sum_{(i,j)\in\mathcal{P}} \left[\sum_{l,l'\in\mathcal{C}:l\neq l'} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) + \left(\sum_{l\in\mathcal{C}} n_{i,j}^{l,l}\right) p(Y=y|L_i=L_j, \boldsymbol{\theta}_{i,j}^k) \right] ,$$

and

$$\left[\boldsymbol{a}_i^{k+1}\right]_{y,l} \propto \sum_{y\in\mathcal{W}:j\neq i} \left[\sum_{l'\in\mathcal{C}:l'\neq l} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) + \left(\sum_{l'\in\mathcal{C}} n_{i,j}^{l',l'}\right) \mathcal{Q}_{i,j}^2(y,0,l|\boldsymbol{\theta}_{i,j}^k) \right] .$$

Here, m_y^{k+1} and $\left[\boldsymbol{a}_i^{k+1}\right]_{y,l}$ should be normalized such that $\sum_{y\in\mathcal{C}} m_y^{k+1} = 1$ and $\sum_{l\in\mathcal{C}} \left[\boldsymbol{a}_i^{k+1}\right]_{y,l} = 1$. The details for deriving the updating equations are shown in Appendix E. To summarize, given an initial estimation of $\boldsymbol{\theta}$, we maximize $\bar{\ell}_p(\boldsymbol{\theta})$ by iteratively applying the above updating equations until its convergence. According to the updating equations, the time complexity and the space complexity for the EM algorithm are provided in the following theorem.

Theorem 6 (Complexity of EM algorithm) *The time complexity for one iteration of the EM algorithm is $O(|\mathcal{P}|C^3)$, where $|\mathcal{P}|$ is the number of worker pairs, and the space complexity is $O(|\mathcal{P}|C^2)$.*

5.2 Maximizing Penalized Pairwise Profile Likelihood

Next, we discuss the algorithm for maximizing $\bar{f}(\boldsymbol{\theta}) = \bar{\ell}_p(\boldsymbol{\theta}) - \sum_{(i,j)\in\mathcal{P}} n_{i,j}\mathcal{J}_\lambda(H_{i,j})$. Since $\bar{f}(\boldsymbol{\theta})$ only differs from $\bar{\ell}_p(\boldsymbol{\theta})$ by a penalty term, an intuitive way for maximizing $\bar{f}(\boldsymbol{\theta})$ is to modify $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$ in the EM algorithm to consider the penalty term. However, $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$ contains the term $\log H_{i,j}$, which means that no matter which penalty is added to $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$, $H_{i,j}$ will never be shrunk to 0. Therefore, the EM algorithm does not help keep the model parsimonious. To tackle this issue, we propose to adopt a coordination descent (CD) algorithm as follows.
Step 1: Update $H_{i,j}$ by maximizing $\bar{f}(\boldsymbol{\theta})$ with $\boldsymbol{a}_i$ and $\boldsymbol{m}$ fixed, i.e.,

$$\boldsymbol{H}^{k+1} = \operatorname*{argmax}_{\boldsymbol{H}} \bar{f}(\boldsymbol{\theta}|\boldsymbol{a}_i = \boldsymbol{a}_i^k, \boldsymbol{m} = \boldsymbol{m}^k) \ ,$$

where $\boldsymbol{H}^{k+1} = \{H_{i,j}^{k+1}, (i,j)\in\mathcal{P}\}$.
Step 2: Update $\boldsymbol{a}_i$ and $\boldsymbol{m}$ by maximizing $\bar{f}(\boldsymbol{\theta})$ with $H_{i,j}$ fixed, i.e.,

$$(\boldsymbol{a}_i^{k+1}, \boldsymbol{m}^{k+1}) = \operatorname*{argmax}_{\boldsymbol{a}_i,\boldsymbol{m}} \bar{f}(\boldsymbol{\theta}|\boldsymbol{H} = \boldsymbol{H}^{k+1}) \ .$$

In Step 1, since $\boldsymbol{a}_i$ and $\boldsymbol{m}$ are fixed, it is straightforward to see that $H_{i,j}$ for each pair of workers can be maximized separately. Therefore, we only need to solve $|\mathcal{P}|$ univariate maximization problems, where $|\mathcal{P}|$ denotes the number of elements in $\mathcal{P}$. In Step 2, since the penalty term is only related to $H_{i,j}$ and $H_{i,j}$ is fixed, maximizing $\bar{f}(\boldsymbol{\theta})$ is equivalent to maximizing $\bar{\ell}_p(\boldsymbol{\theta})$. Consequently, we can directly adopt the EM algorithm without updating $H_{i,j}$ to derive $\boldsymbol{a}_i$ and $\boldsymbol{m}$. In practice, one possible modification of Step 2 is to consider the likelihood for non-colluding worker pairs instead of the profile likelihood. As mentioned before, in the penalized pairwise profile likelihood method, we eliminate the parameters $\boldsymbol{b}_{i,j}$ at a cost of discarding some information. Thus, for a pair of non-colluding workers with $H_{i,j} = 0$, the profile likelihood does not fully explore all the available information and may lead to multiple maximizers of $\bar{\ell}_p(\boldsymbol{\theta})$. To address this issue, given the estimated result of $\boldsymbol{H}$ in Step 1, we propose to maximize the following objective function in Step 2

$$g(\boldsymbol{\theta}|\boldsymbol{H} = \boldsymbol{H}^k) = \sum_{(i,j)\in\mathcal{P}:H_{i,j}=0} \ell_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) + \sum_{(i,j)\in\mathcal{P}:H_{i,j}\neq 0} \bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) \ .$$

In other words, we consider profile likelihood for colluding workers and consider likelihood for non-colluding workers. As a result, the updating equations for $\boldsymbol{a}_i$ and $\boldsymbol{m}$ change to

$$m_y^{k+1} \propto \sum_{(i,j)\in\mathcal{P}:H_{i,j}=0} \sum_{l,l'\in\mathcal{C}} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) + \sum_{(i,j)\in\mathcal{P}:H_{i,j}\neq 0} \left[\sum_{l,l'\in\mathcal{C}:l\neq l'} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) + \left(\sum_{l\in\mathcal{C}} n_{i,j}^{l,l} \right) p(Y=y|L_i=L_j,\boldsymbol{\theta}_{i,j}^k) \right] ,$$

and

$$\left[\boldsymbol{a}_i^{k+1}\right]_{y,l} \propto \sum_{j\in\mathcal{W}:H_{i,j}=0} \sum_{l'\in\mathcal{C}} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) + \sum_{j\in\mathcal{W}:H_{i,j}\neq 0} \left[\sum_{l'\in\mathcal{C}:l'\neq l} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) + \left(\sum_{l'\in\mathcal{C}} n_{i,j}^{l',l'} \right) \mathcal{Q}_{i,j}^2(y,0,l|\boldsymbol{\theta}_{i,j}^k) \right] .$$

The detailed derivation is similar as in Appendix E, and thus is omitted.

6. Ground Truth Inference

After obtaining the estimated parameter $\hat{\boldsymbol{\theta}}$, we can infer the ground true label y_t for each task. Since we adopt the pairwise likelihood for parameter estimation, we consider the composite posterior distribution based on the pairwise likelihood. The composite posterior distribution is similar to the posterior distribution, but with the full likelihood replaced by the composite likelihood, which is the pairwise likelihood in our case (Ribatet et al., 2012). Specifically, the composite posterior distribution of y_t can be calculated as

$$\pi(y_t=c|l_{i,t},i\in\mathcal{W}) \propto \hat{m}_c \prod_{(i,j)\in\mathcal{P}:l_{i,t}\neq 0,l_{j,t}\neq 0} p(L_i=l_{i,t},L_j=l_{j,t}|y_t=c,\hat{\boldsymbol{\theta}}) .$$

Here the conditional probability is

$$p(L_i=l_{i,t},L_j=l_{j,t}|y_t=c,\hat{\boldsymbol{\theta}}) = [\hat{\boldsymbol{a}}_i]_{c,l_{i,t}}[\hat{\boldsymbol{a}}_j]_{c,l_{j,t}}(1-\hat{H}_{i,j}) + I(l_{i,t}=l_{j,t})p(L_i=L_j=l_{i,t}|Z_{i,j}=1,y_t=c)\hat{H}_{i,j} .$$

The probability $p(L_i=L_j=l_{i,t}|Z_{i,j}=1,y_t=c)=[\boldsymbol{b}_{i,j}]_{c,l_{i,t}}$ is unknown, because $\boldsymbol{b}_{i,j}$ has been eliminated from the model. To address this problem, we propose to consider a non-informative Dirichlet prior for $\boldsymbol{b}_{i,j}$ and calculate the expected composite posterior $\pi(y_t=c|l_{i,t},i\in\mathcal{W})$. Specifically, let the prior distribution for each row of $\boldsymbol{b}_{i,j}$ be

$$[\boldsymbol{b}_{i,j}]_c \sim \text{Dir}\left(\frac{1}{C},\cdots,\frac{1}{C}\right) ,$$

and assume $\boldsymbol{b}_{i,j}$ to be mutually independent. The expected composite posterior is

$$
\begin{aligned}
&E_{\{\boldsymbol{b}_{i,j},(i,j)\in\mathcal{P}\}}\left[\pi(y_t = c|l_{i,t}, i \in \mathcal{W})\right] \\
&\qquad \propto \hat{m}_c \prod_{(i,j)\in\mathcal{P}:l_{i,t}\neq 0, l_{j,t}\neq 0} E_{\boldsymbol{b}_{i,j}}\left[p(L_i = l_{i,t}, L_j = l_{j,t}|y_t = c, \hat{\boldsymbol{\theta}})\right] \\
&\qquad = \hat{m}_c \prod_{(i,j)\in\mathcal{P}:l_{i,t}\neq 0, l_{j,t}\neq 0} \{[\hat{\boldsymbol{a}}_i]_{c,l_{i,t}}[\hat{\boldsymbol{a}}_i]_{c,l_{j,t}}(1 - \hat{H}_{i,j}) + \frac{1}{C} I(l_{i,t} = l_{j,t})\hat{H}_{i,j}\} \ .
\end{aligned}
$$

As a result, by considering the expected composite posterior, we essentially replace the probability $p(L_i = L_j = l_{i,t}|Z_{i,j} = 1, y_t = c)$ by $1/C$. In other words, we treat workers as spammers when they collude on a task, which is equivalent to removing the duplicated labels generated by colluding workers. With a greater value of $\hat{H}_{i,j}$, the probability $p(L_i = l_{i,t}, L_j = l_{j,t}|y_t = c, \hat{\boldsymbol{\theta}})$ gets closer to $1/C$ and thus has less effect on the final inference result. In this way, we discard the information from collusion to protect the final inference result from over-weighting the duplicated labels generated by colluding workers.

7. Related Work

In the literature, a number of studies have been focused on ground truth inference in crowdsourcing. One of the most commonly used models is the General DS model (Dawid and Skene, 1979), which is a baseline model of our study. As mentioned before, the General DS model is flexible in modeling worker expertise and can reduce to several other models by imposing some restrictions. Extensions have also been made to incorporate more factors into consideration. For examples, Donmez et al. (2010) considered the workers' accuracy changed over time, and adopted a particle filter to trace the change. Liu et al. (2012) transformed this problem into an inference problem in graphical models and applied approximate variational methods for ground truth inference. Whitehill et al. (2009) assumed different tasks had different levels of difficulty, and Kamar et al. (2015) further discussed different hierarchical models to account for task-dependent bias, where the confusion matrix was dependent on the task feature. Another perspective in crowdsourcing is to train a supervised model based on the collected labels. Raykar et al. (2010) discussed supervised model training for different data types of response with noisy labels, while the worker expertise was still considered as a confusion matrix. Other studies extended the method and assumed each worker's response was an output of a different supervised model (Yan et al., 2010; Welinder et al., 2010; Kajino et al., 2012; Bi et al., 2014).

To further improve the performance of the ground truth inference algorithm, Raykar and Yu (2012) proposed to explicitly eliminate spammers by designing a special prior distribution of workers' confusion matrices. However, this model regards the workers as independent and only considers individual spam. Jagabathula et al. (2017) proposed a reputation algorithm to identify unreliable and adversarial workers without a probabilistic labeling strategy. Some other studies focused on detecting the community structure of workers (Kajino et al., 2013; Venanzi et al., 2014; Moreno et al., 2015). Specifically, workers were assigned to different communities where within the same community, workers shared the same or similar expertise. These methods still assumed the workers to be independent.

Only a few studies have considered the collusive behaviors of workers. Ghosh et al. (2011), Karger et al. (2013), and Jagabathula et al. (2017) considered the case when adversarial workers colluded to attack the system, but they mainly focused on analyzing the robustness of a model against the attack in the worst case and did not consider other collusive behaviors. Liu et al. (2008), Xu (2013), and Allahbakhsh et al. (2013) discussed collusion detection in reputation and rating systems. These studies extracted features from the ratings of workers and then identified colluding groups based on a classifier or a clustering algorithm. For crowdsourcing systems with labeling tasks, Chen et al. (2018) proposed an algorithm to remove duplicated labels resulted from worker collusion before aggregating the labels. However, the objective function of the algorithm is a heuristic metric and thus the performance still needs to be verified.

To summarize, most existing studies assume the workers to be independent and only a few studies have considered the collusive behavior of workers. To the best of our knowledge, the proposed PROCAP method in this paper is the first rigorous statistical model that aims to solve collusion detection and ground truth inference with theoretical justifications.

8. Experiments

We conduct a series of numerical studies to verify the performance of our PROCAP method. At first, we simulate the collected labels in different scenarios, and for each scenario, we assess the performance of PROCAP in ground truth inference, collusion detection, and parameter estimation, based on different baseline models. Then, we implement PROCAP in some publicly available real data sets for ground truth inference and collusion detection.

For ground truth inference, besides PROCAP, we also report the performance of two benchmark methods, including (i) the baseline model without considering worker collusion and (ii) majority voting. In particular, by assuming workers to be independent, a baseline model can be directly used for ground truth inference, which we refer to as independent-worker baseline model for simplicity. The parameters $\boldsymbol{m}$ and $\boldsymbol{a}_i$ can be estimated by maximum likelihood using the EM algorithm, and then the posterior distribution of y_t can be calculated as

$$p(y_t = c | l_{i,t}, i \in \mathcal{W}) \propto \prod_{i \in \mathcal{W}} p(l_{i,t} | y_t = c, \boldsymbol{a}_i) m_c = m_c \prod_{i \in \mathcal{W}} [\boldsymbol{a}_i]_{c, l_{i,t}} \ . \tag{6}$$

Consequently, the estimated label is

$$\hat{y}_t = \underset{c \in \mathcal{C}}{\operatorname{argmax}} \, p(y_t = c | l_{i,t}, i \in \mathcal{W}) \ . \tag{7}$$

The details of independent-worker baseline models can be found in Dawid and Skene (1979). It is worth noting that for the Uniform DS model, (6) reduces to

$$p(y_t = c | l_{i,t}, i \in \mathcal{W}) \propto a^{\sum_i I(l_{i,t}=c)} \left(\frac{1-a}{C-1} \right)^{\sum_i [1 - I(l_{i,t}=c)]} m_c \ .$$

If we further restrict $m_c = 1/C$ and $a > 1/C$, the estimated $\hat{y}_t$ using (7) is the same as the one in (8) based on majority voting:

$$\hat{y}_t = \underset{y \in \mathcal{C}}{\operatorname{argmax}} \sum_i I(l_{i,t} = y) \ , \tag{8}$$

which shows the connection between the Uniform DS model and majority voting.

In the literature of crowdsourcing, the algorithms for estimating the parameters of the independent-worker baseline models are usually initialized by majority voting (Dawid and Skene, 1979). Specifically, this is equivalent to setting the initial estimation of the parameters as

$$\left[\boldsymbol{a}_i^0\right]_{y,l} = \begin{cases} 0.7\ , & \text{if } y = l\ , \\ \frac{0.3}{C-1}\ , & \text{if } y \neq l\ , \end{cases} \tag{9}$$

and $m_c^0 = 1/C$. Therefore, throughout this section, we initialize $\boldsymbol{a}_i$ and $\boldsymbol{m}$ in the same way when implementing the algorithms proposed in Section 5. For $H_{i,j}$, we initialize $H_{i,j}^0 = 0.5$. The numerical studies were conducted on a virtual machine with an Intel Xeon E5-2693V3 16-core 2.30-GHz processor and 32 GB RAM.

8.1 Synthetic Data Sets

In the simulation study, we explore the performance of our proposed PROCAP method under different scenarios and compare with the benchmark methods. While the method can be similarly applied to multi-class labeling tasks, we consider tasks of two alternatives, i.e., $C = 2$, in this simulation study to numerically demonstrate the performance and properties of the method. Additional simulation studies with $C > 2$ have been conducted and similar results are obtained, and thus are omitted in this paper. The ground true labels y_t for each task are randomly generated with the marginal probability $\boldsymbol{m}^* = [0.6, 0.4]^T$. In addition, we consider 10 workers. If working independently, each worker has a confusion matrix of $\boldsymbol{a}^* = \begin{bmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{bmatrix}$. The first k workers belong to a colluding group with a colluding probability of h for each task. Specifically, with a probability of h, the k workers collude on a task, and they generate the same label according to a confusion matrix of $\boldsymbol{b}^* = \begin{bmatrix} \rho & 1-\rho \\ 1-\rho & \rho \end{bmatrix}$, otherwise they generate the labels independently according to their own confusion matrices. The remaining $10 - k$ workers are always independent. Throughout the simulation, each worker generates labels for all tasks.

In the simulation study, different scenarios with different combinations of group size k and colluder expertise ρ are considered. Specifically, we consider scenarios with $k = 3$ and $k = 5$, representing the number of colluding workers to be small and large, respectively; and we consider $\rho = 0.7, 0.5, 0.3$, and 0, representing the colluding workers to be reliable workers (tend to give the correct label), spammers (randomly generate labels), adversarial workers (tend to give a wrong label), and sophisticated adversarial workers (always give wrong labels), respectively. Table 1 summarizes the scenarios that we consider in the simulation study. In each scenario, we consider two different colluding probabilities including $h = 0.5$ and $h = 1$, representing the colluding workers collude on half of the tasks and all the tasks, respectively. In addition, we try different numbers of tasks to see how the performance of PROCAP changes. For each scenario with a certain number of tasks, we replicate the simulation for 100 times.

We consider two baseline models including the Uniform and the Homogeneous DS model. The result of each scenario is discussed as follows.

Group size k	Colluder expertise ρ			
	$\rho = 0.7$ (reliable workers)	$\rho = 0.5$ (spammers)	$\rho = 0.3$ (adversarial workers)	$\rho = 0$ (sophisticated adversarial workers)
$k = 3$ (small group)	Scenario 1	Scenario 2	Scenario 3	Scenario 4
$k = 5$ (large group)	Scenario 5	Scenario 6	Scenario 7	Scenario 8

Table 1: Summary of scenarios considered in the simulation study.

Scenario 1: small group with honest workers

In this scenario, we consider $k = 3$ and $\rho = 0.7$. Since now $\boldsymbol{b}^* = \boldsymbol{a}^*$, the *non-colluding assumption* is satisfied. In addition, the first three workers have an overall confusion matrix of $(1-h)\boldsymbol{a}^* + h\boldsymbol{b}^*$ that is equal to the overall confusion matrix $\boldsymbol{a}^*$ for the other workers. This satisfies the assumption of the independent-worker Uniform DS model and majority voting that the confusion matrix of each worker is the same. At first, we examine the accuracy of ground truth inference as shown in Figure 1. This figure is divided into four panels, where the two columns represent the two baseline models, the Uniform and Homogeneous DS models, and the two rows represent two different cases with $h = 0.5$ and $h = 1$, respectively. If $h = 0.5$, it means that the three workers collude on roughly half of the tasks, but for the other tasks, they give their honest answers independently. If $h = 1$, it means that these workers always collude and give the same label. For each panel, we compare the accuracy of the majority voting model, the independent-worker baseline model (Independent), and the proposed PROCAP model with collusion detection based on the baseline model with the tuning parameter λ selected by AIC and BIC (Collusion-AIC and Collusion-BIC), for different number of tasks. Since majority voting (see (8)) is independent of the baseline model, for a certain h, it has the same curve in the left panel and the right panel.

From Figure 1, we have the following observations. First, with the Uniform DS model as baseline, PROCAP performs better than the other two benchmarks that do not consider worker collusion, and the difference becomes greater with a larger h. This indicates that by detecting the worker collusion, PROCAP can improve the accuracy of ground truth inference. Second, with the Homogeneous DS model as the baseline, the performance of PROCAP increases with more tasks. This is because in this case, there are relatively more parameters that need to be estimated, and with more tasks, the estimation becomes more accurate, leading to a higher accuracy in ground truth inference. Last, we can see that the independent-worker Homogeneous DS model does not perform well. When workers collude and frequently generate the same label, it tends to regard the colluding workers as experts who seldom make mistakes, which leads to a biased estimation of the workers' confusion matrices. In comparison, the performance of majority voting and the independent-worker Uniform DS model is not too bad. The reason is that for these two models, the confusion matrix of each worker is restricted to be the same, which avoids incorrectly treating colluding workers as experts as in the independent-worker Homogeneous DS model.

Besides the performance in ground truth inference, we are also interested in the performance of parameter estimation and collusion detection of PROCAP. Here we only consider the case $h = 0.5$ and the estimation selected by BIC. To quantify the performance of pa-

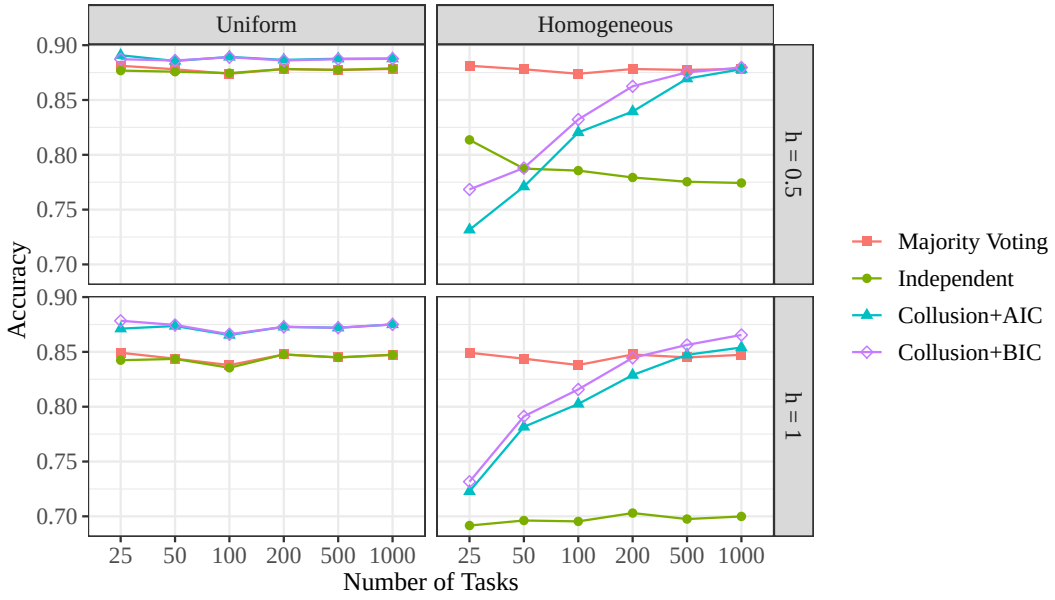

Figure 1: Performance of ground truth inference using different models for Scenario 1.

rameter estimation, we calculate the L_1 distance between the estimated $\hat{H}_{i,j}$ and the true $H^*_{i,j}$ by $\sum_{(i,j)\in\mathcal{P}} |\hat{H}_{i,j} - H^*_{i,j}|$. In Figure 2, we show the average and one standard deviation of the L_1 distance over 100 repetitions based on the two baseline models. As we can see, for both baseline models, the L_1 distance diminishes to 0, and we achieve relatively smaller distance based on the Uniform DS model, because less parameters need to be estimated.

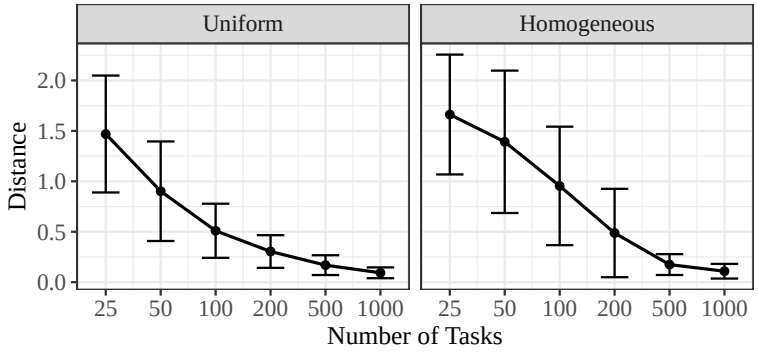

Figure 2: L_1 distance between the the estimated $\hat{H}_{i,j}$ and the true $H^*_{i,j}$ in Scenario 1.

In Figure 3, we further show the probability that our method correctly detects the colluding workers, i.e., $p(\hat{\mathcal{P}}_0 = \mathcal{P}^*_0)$, which is calculated as the proportion of repetitions where the set of detected colluding worker pairs $\hat{\mathcal{P}}_0$ equals the set of true colluding worker pairs $\mathcal{P}^*_0$ among 100 repetitions. Obviously, the probability increases with more tasks for both baseline models.

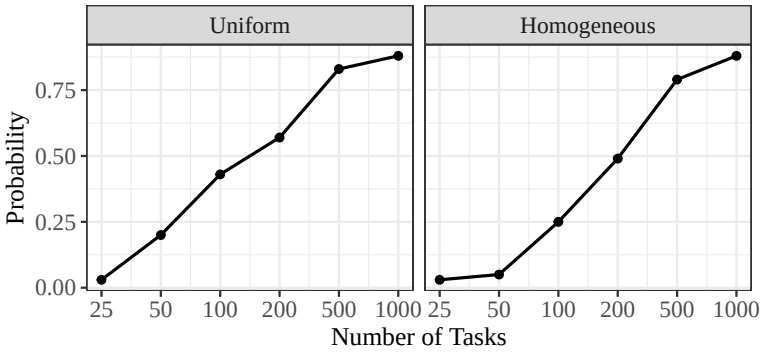

Figure 3: Probability of selecting the correct model in Scenario 1.

The running time of the proposed PROCAP method was recorded for the synthetic data sets. Table 2 shows the average running time until convergence in seconds based on a single CPU core for the EM algorithm described in Section 5.1 and the coordinate descent (CD) algorithm described in Section 5.2 under Scenario 1 with $h = 0.5$, where the EM algorithm (or the CD algorithm) stops when the log-likelihood $\bar{\ell}_p(\boldsymbol{\theta})$ (or the penalized log-likelihood $\bar{f}(\boldsymbol{\theta})$) decreases by a proportion of less than 10^{-6} after the latest iteration. The numbers in parentheses are the average number of iterations for the algorithms to converge. It is easy to see that a larger number of tasks does not necessarily increase the running time, because only the summary statistic $n_{i,j}^{l,l'}$ is needed in the PROCAP method as in (3). In fact, EM algorithm converges in fewer iterations with more tasks, and thus leading to less computation time. For the CD algorithm, the number of iterations seems to be stable under the Uniform DS baseline model and increases with more tasks under the Homogeneous DS baseline model. However, since each iteration of the CD algorithm contains a run of the EM algorithm and the running time of the EM algorithm decreases with more tasks, the running time of the CD algorithm also decreases with more tasks in this example.

		Number of Tasks					
		25	50	100	200	500	1000
EM	Uniform	0.083	0.065	0.046	0.037	0.032	0.028
		(11.0)	(8.7)	(5.8)	(4.5)	(3.7)	(3.1)
	Homogeneous	0.202	0.124	0.085	0.064	0.045	0.040
		(28.6)	(17.5)	(11.9)	(8.8)	(6.2)	(5.5)
CD	Uniform	0.425	0.378	0.300	0.244	0.185	0.140
		(4.7)	(5.2)	(5.6)	(5.5)	(4.7)	(4.0)
	Homogeneous	1.165	1.011	1.013	0.957	0.819	0.729
		(5.6)	(7.7)	(11.0)	(13.6)	(16.4)	(16.3)

Table 2: Computation cost for Scenario 1 with $h = 0.5$.

Scenario 2: small group with spammers

In this scenario, we consider $k = 3$, $\rho = 0.5$, i.e., when the first three workers collude on a task, they simply give a random guess. In this case, $\boldsymbol{a}^* \neq \boldsymbol{b}^*$ and thus when $h = 0.5$, the *non-colluding assumption* is violated. In addition, the majority voting and the independent-worker Uniform DS model may not be a good fit, because the overall confusion matrix $(1-h)\boldsymbol{a}^* + h\boldsymbol{b}^*$ for the first three workers is different from the overall confusion matrix $\boldsymbol{a}^*$ for other workers, which contradicts the assumption of equal confusion matrices in these two models. In Figure 4, we show the performance of ground truth inference by different methods. Similar conclusions can be drawn as in Scenario 1. PROCAP performs the

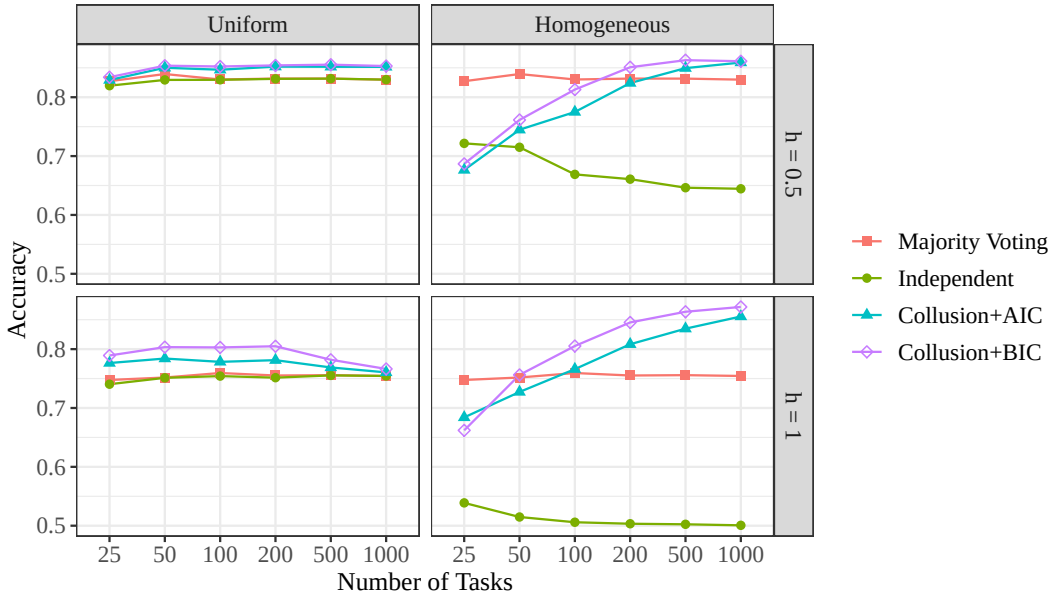

Figure 4: Performance of ground truth inference using different models for Scenario 2.

best among all methods. The performance of majority voting and the independent-worker Uniform DS model is not too bad, probably because $\boldsymbol{b}^*$ is still close to $\boldsymbol{a}^*$. However, the independent-worker Homogeneous DS model is vulnerable to worker collusion with the worst performance. A new observation is that compared with Figure 1, the performance of PROCAP based on the Homogeneous DS model is much better than that based on the Uniform DS model when a large number of tasks are available. The reason is that in this scenario, the Homogeneous DS model provides a better fit to the data. Specifically, as discussed in Section 3.2, with a higher value of colluding probability h for the first three workers, the *non-colluding assumption* provides a good approximation only if we can regard the confusion matrices $\boldsymbol{a}_i, i = 1, 2, 3$ for the first three workers to be close to $\boldsymbol{b}^*$. However, this is not satisfied in the Uniform DS model where $\boldsymbol{a}_i$ for all workers are restricted to be the same. By combining the worker collusion detection and the flexibility of the Homogeneous DS model, PROCAP achieves superior performance, when there are a large number of tasks available.

Similar as before, we check the performance of PROCAP in parameter estimation and collusion detection for $h = 0.5$ with the parameter selected by the BIC criterion. The results are shown in Figure 5 and Figure 6. It is interesting to see that with the Uniform DS model as the baseline, for 1000 tasks, the standard deviation of the L_1 distance between $\hat{H}_{i,j}$ and $H^*_{i,j}$ increases, and the probability of correctly detecting colluding workers decreases. This is because the *non-colluding assumption* cannot provide a good approximation when the Uniform DS model is the baseline, as mentioned before. On contrary, based on the Homogeneous DS model, although the *non-colluding assumption* still does not strictly hold, PROCAP shows good performance in parameter estimation and collusion detection. From the result, we can see that PROCAP is insensitive to this assumption if a proper baseline model is used.

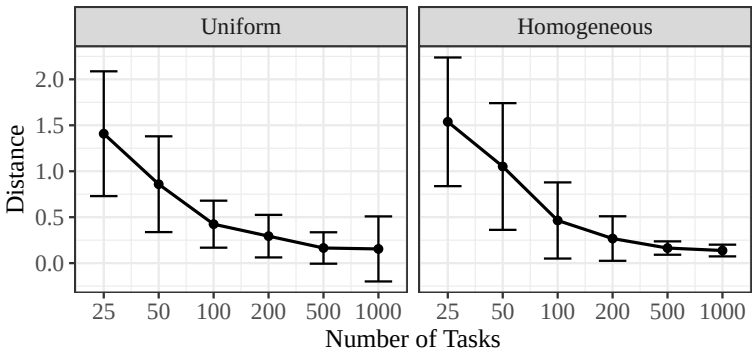

Figure 5: L_1 distance between the estimated $\hat{H}_{i,j}$ and the true $H^*_{i,j}$ in Scenario 2.

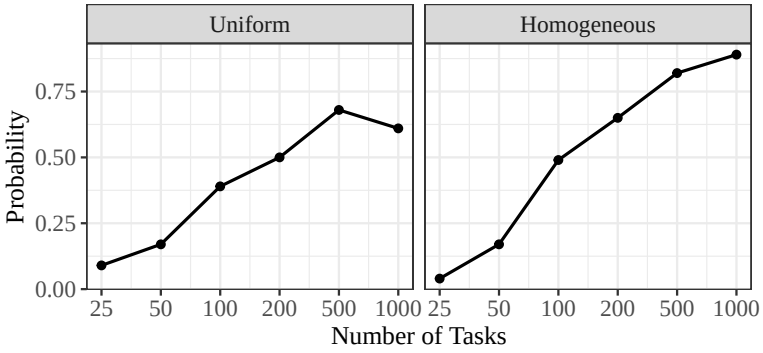

Figure 6: Probability of selecting the correct model in Scenario 2.

Scenario 3: small group with adversarial workers

In this scenario, we consider $k = 3$ and $\rho = 0.3$. In this case, the colluding workers try to select the wrong answer to mislead the inference when they collude. Similar as before, we show the result of ground truth inference in Figure 7. As this figure shows, the performance of independent-worker Uniform DS model and majority voting deteriorates because the

overall confusion matrix of the first three workers greatly differ from that of the other workers. In addition, when the Uniform DS model is used as the baseline, PROCAP does not perform much better than the benchmark models. In fact, it is even worse than the benchmark models when $h = 1$ and the number of tasks is large. The reason is similar as in Scenario 2 that the *non-colluding assumption* does not provide a good approximation under the constraint of equal confusion matrices by the Uniform DS model. Since $\boldsymbol{b}^*$ much differs from $\boldsymbol{a}^*$ in this scenario, the influence of the improper baseline model becomes more severe. In contrast, the performance of PROCAP is satisfactory when the Homogeneous DS model is used as the baseline with a large number of tasks. Although the *non-colluding assumption* still does not strictly hold in this scenario, the Homogeneous DS model provides a better fit and leads to higher accuracy in ground truth inference when incorporated by PROCAP. In addition, to achieve a good performance, PROCAP requires relatively large number of tasks as many parameters are involved in the Homogeneous DS model. Therefore, it is important to select a suitable baseline model. If the baseline model is oversimplified and does not fit the data well, or the baseline model is too complicated but the available data is not enough to provide a good estimation of the parameters, PROCAP may not perform well.

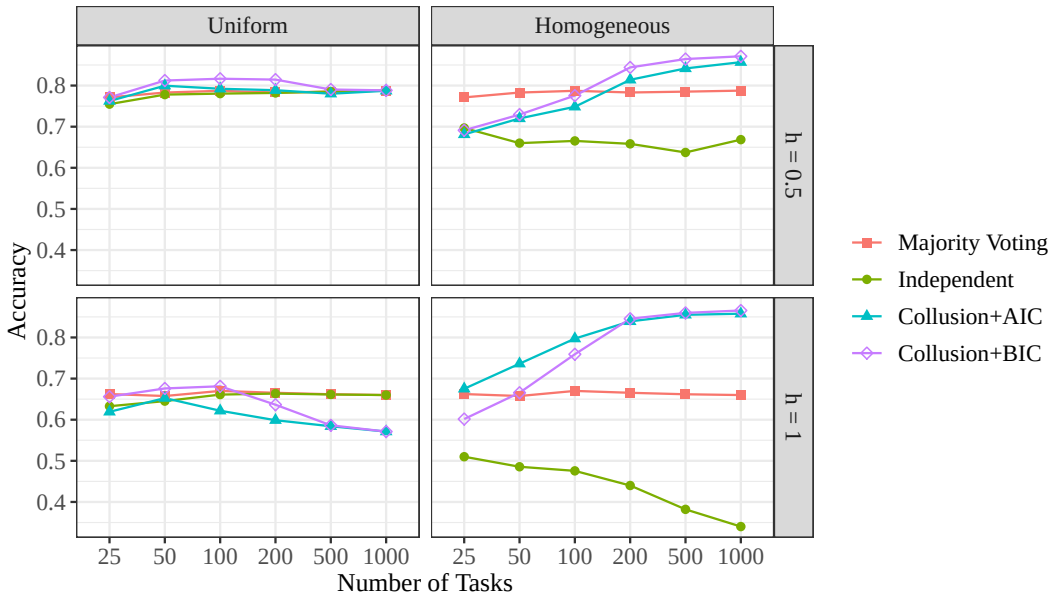

Figure 7: Performance of ground truth inference using different models for Scenario 3.

Next, we show the parameter estimation performance of PROCAP using BIC when $h = 0.5$ as in Figure 8 and Figure 9. It is clear that with the Homogeneous DS model as the baseline, the performance of PROCAP becomes better with more tasks, and if the number of tasks is large enough, we can still obtain an accurate estimation with a high probability of detecting the correct colluding workers. However, it is not true when the Uniform DS model is used as the baseline. Similar as before, the reason is because the Uniform DS model is too restricted and cannot provide a good fit to the data, leading to a biased estimation.

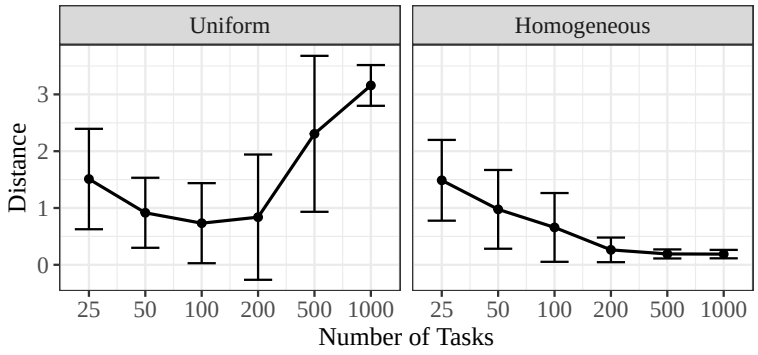

Figure 8: L_1 distance between the estimated $\hat{H}_{i,j}$ and the true $H^*_{i,j}$ in Scenario 3.

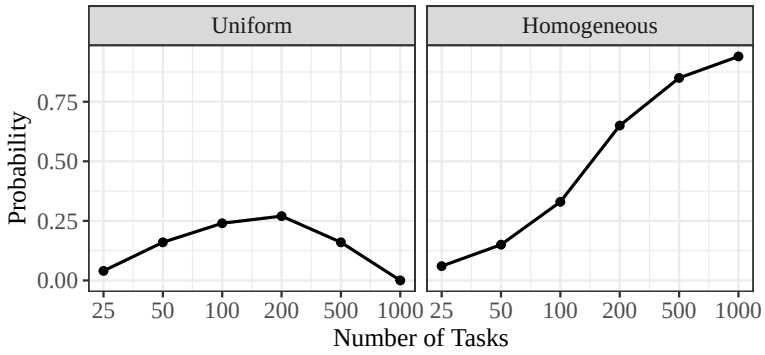

Figure 9: Probability of selecting the correct model in Scenario 3.

Scenario 4: small group with sophisticated adversarial workers

In this scenario, we consider $k = 3$ and $\rho = 0$. In other words, when the workers collude, they always submit the wrong answer. Here a sophisticated adversarial worker refers to a worker who knows the correct answer but intentionally gives the wrong answer. The result of ground truth inference is shown in Figure 10. Similar to Scenario 3, when $h = 0.5$, the performance of PROCAP is very good if the Homogeneous DS model is the baseline, but degrades when the Uniform DS model is the baseline. Interestingly, when $h = 1$, the performance of majority voting deteriorates much but the independent-worker Homogeneous DS model achieves very good accuracy. The reason is that in this case, the three colluding workers always generate the wrong answer and can simply be regarded as independent adversarial workers. Therefore, the independent-worker Homogeneous DS model can correctly estimate their confusion matrices. By comparing the case $h = 0.5$ with $h = 1$, we can see that the independent-worker Homogeneous DS model possesses the ability to identify a small number of sophisticated adversarial workers when they always give the wrong answer, but if they try to disguise to be honest workers by honestly answering a subset of the tasks, then the model will fail to identify them. We have also checked the performance of parameter estimation for PROCAP in this scenario. The result is similar to Scenario 3 and thus is omitted.

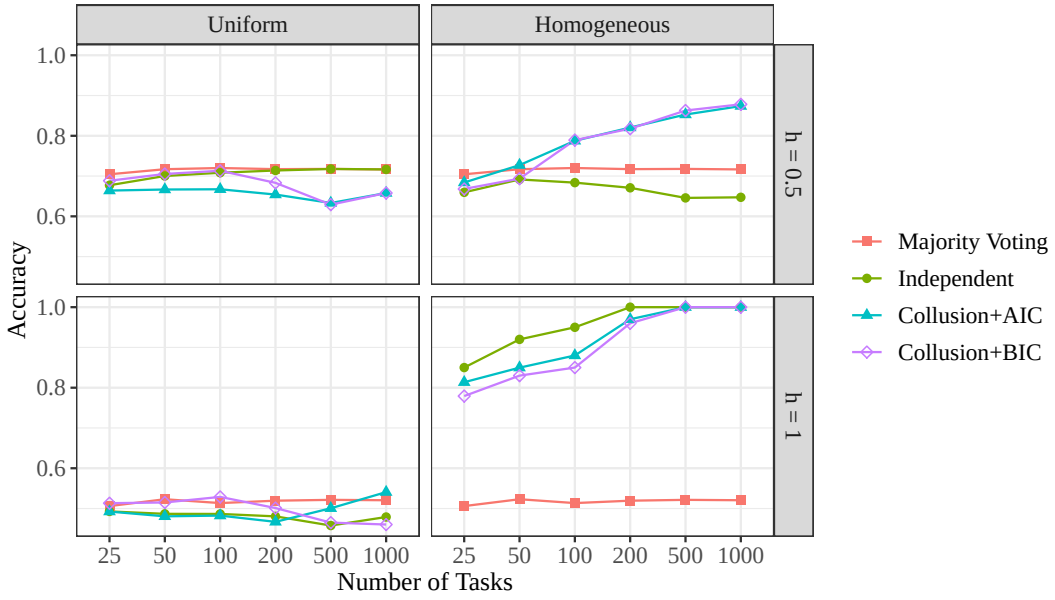

Figure 10: Performance of ground truth inference using different models for Scenario 4.

Scenario 5: large group with honest workers

In this scenario, we consider $k = 5$ and $\rho = 0.7$. Here a half of the workers are involved in collusion. The result of ground truth inference is shown in Figure 11. Compared with Scenario 1, the performance of majority voting greatly deteriorates, especially for $h = 1$, because more workers are colluding with others. In contrast, PROCAP still achieves high accuracy in ground truth inference.

Scenario 6: large group with spammers

In this scenario, we consider $k = 5$ and $\rho = 0.5$. The result of ground truth inference is shown in Figure 12. We can see that the accuracy of majority voting and the independent-worker baseline models is close to 0.5 when $h = 1$, but the accuracy of PROCAP can reach above 0.8 when a large number of tasks are available, demonstrating great superiority.

Scenario 7: large group with adversarial workers

In this scenario, we consider $k = 5$ and $\rho = 0.3$, and the result of ground truth inference is shown in Figure 13. Since a half of the workers are colluding and adversarial, the accuracy of majority voting and independent-worker baseline models can drop below 0.5, meaning that they are worse than random guess. In contrast, the accuracy of PROCAP is still much better when the Uniform DS model is the baseline, and can even achieve 0.8 with enough number of tasks when the Homogeneous DS model is the baseline.

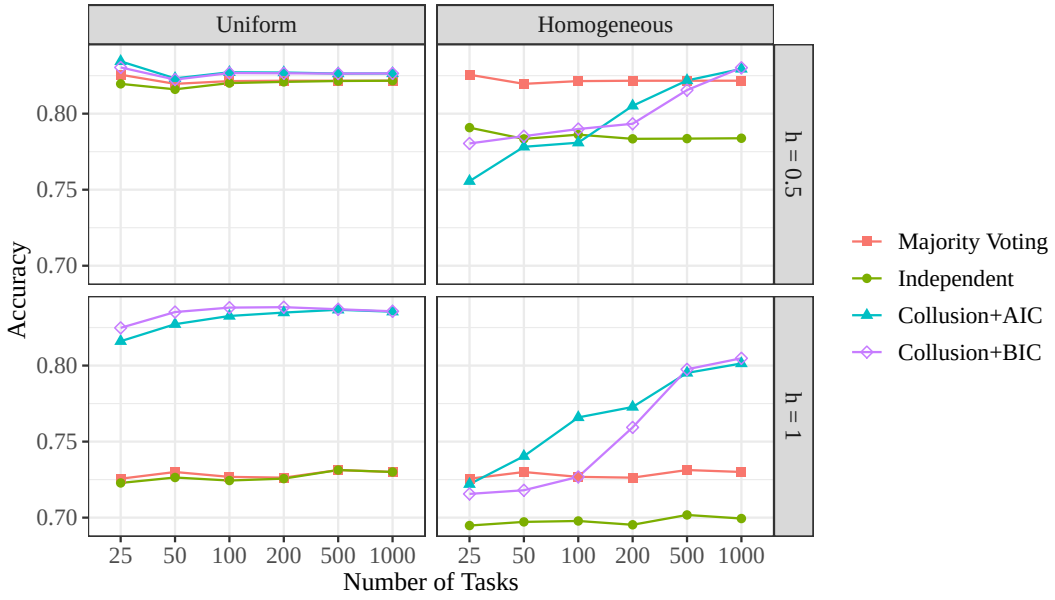

Figure 11: Performance of ground truth inference using different models for Scenario 5.

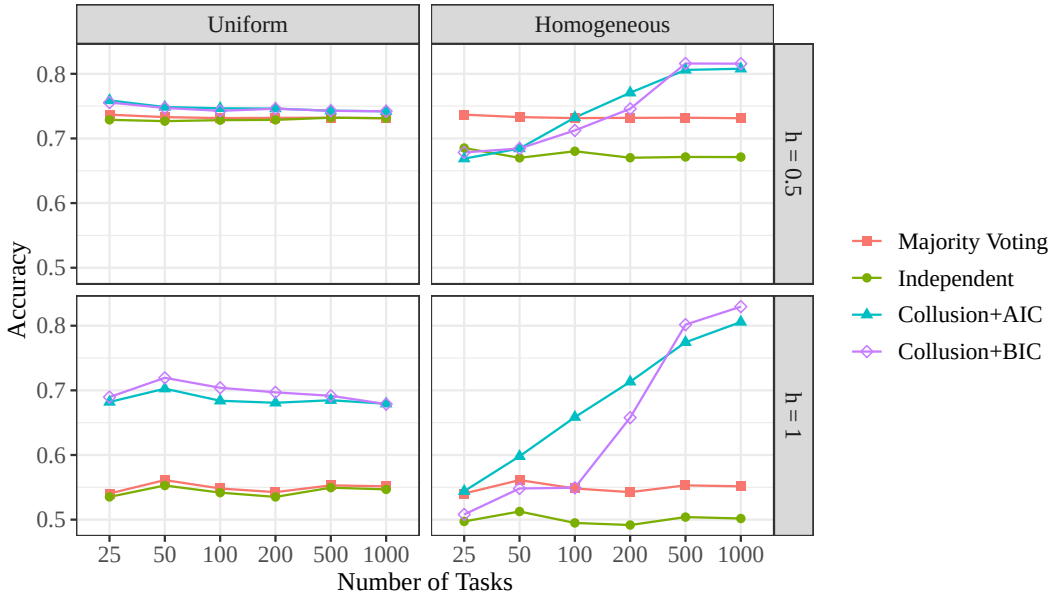

Figure 12: Performance of ground truth inference using different models for Scenario 6.

Scenario 8: large group with sophisticated adversarial workers

At last, we consider the scenario with $k = 5$ and $\rho = 0$. The result of ground truth inference is shown in Figure 14. When $h = 0.5$, we observe relatively good performance of PROCAP.

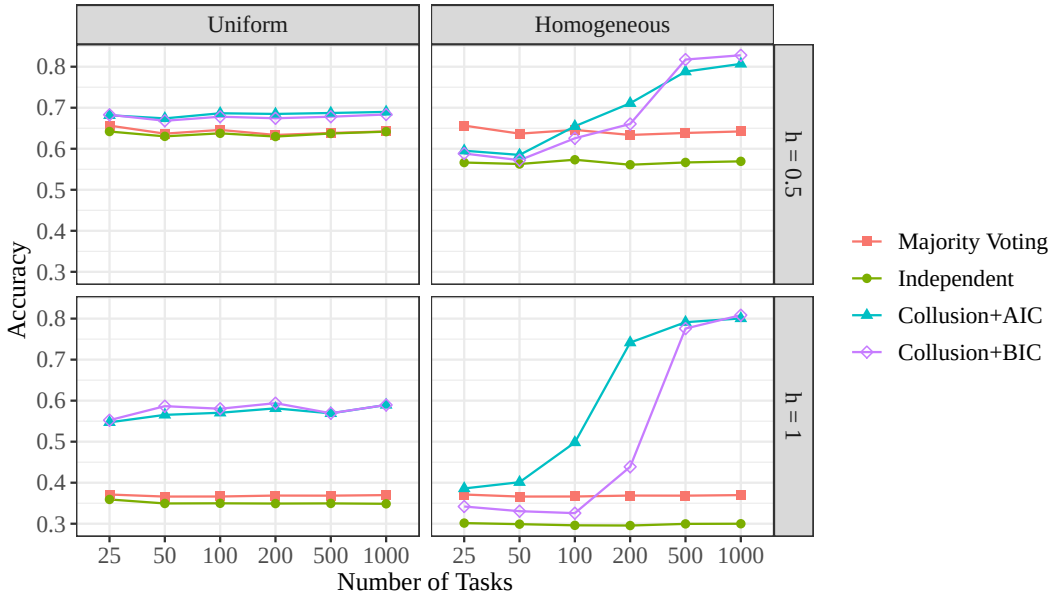

Figure 13: Performance of ground truth inference using different models for Scenario 7.

This means that if the sophisticated adversarial workers try to disguise to be honest workers for a subset of the tasks, PROCAP can accurately identify them given enough tasks.

In contrast, when $h = 1$, the colluding workers can be simply regarded as independent and thus PROCAP delivers similar result as the independent-worker baseline models, which has also been seen in Scenario 4. The bad performance of the independent-worker baseline models and PROCAP when $h = 1$ can be explained by the unidentifiability of the true parameter. As mentioned in Section 4, the true parameter $\boldsymbol{\theta}^*$ may not be identifiable. Since the algorithm for parameter estimation is initialized by (9) that assumes the majority of workers to be honest, the resulting estimated parameter turns out to be biased and has similar likelihood as the true parameter. However, if prior knowledge is available which suggests the majority of workers are adversarial, we can set the initial point for the algorithm as

$$\left[\boldsymbol{a}_i^0\right]_{y,l} = \begin{cases} 0.3, & \text{if } y = l \ , \\ \frac{0.7}{C-1}, & \text{if } y \neq l \ , \end{cases}$$

and then the parameter estimation and ground truth inference will be much more accurate. Therefore, without a suitable initial point or some empirical knowledge that solves the identifiability issue, PROCAP may fail to identify a large group of sophisticated adversarial workers when they always collude, which is the same as the independent-worker baseline model.

To summarize, when workers collude, PROCAP can generally achieve better performance in ground truth inference than the independent-worker baseline models, and detect the worker collusion correctly. In addition, we would like to highlight some points as follows. First, it is important to select a suitable baseline model. If the baseline model is too restrictive and the *non-colluding assumption* cannot provide a good approximation, PRO-

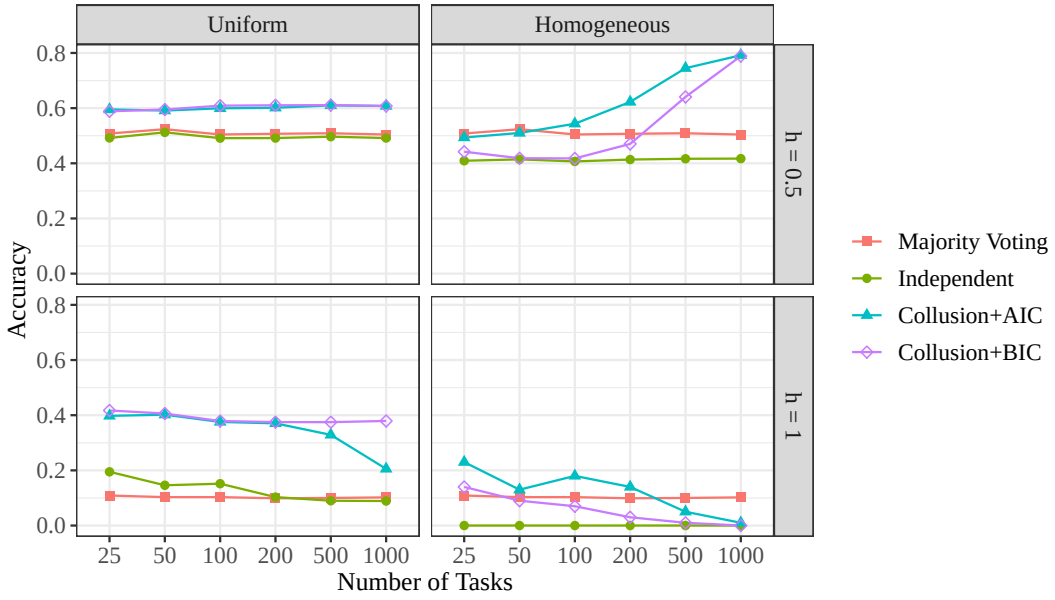

Figure 14: Performance of ground truth inference using different models for Scenario 8.

CAP may not deliver a satisfactory result in ground truth inference. Second, a baseline model that is more flexible with more parameters to be estimated usually requires more data to ensure an accurate estimation. Last, PROCAP is generally more robust to colluding spammers and adversarial workers than the independent-worker baseline models, but when a large number of sophisticated workers are involved and always generate the wrong labels, PROCAP might fail. However, this case is very unlikely to occur in practice.

8.2 Real Data Sets

In addition, we implement PROCAP to five publicly available data sets including *bluebird*, *ducks*, *tweets*, *stage2*, and *rating*. These data sets are briefly described as follows.

(1) The *bluebird* data set consists of worker-generated labels indicating whether an image contains Indigo Bunting or Blue Grosbeak (Welinder et al., 2010);

(2) In the *ducks* data set, workers are presented with photos that may contain American Black Duck, Canada Goose, Mallard, Red-necked Grebe, or no bird, and need to identify whether the photo contains a duck or not (Welinder et al., 2010). Only Mallards and American Black Ducks are ducks;

(3) In the *tweets* data set, workers classify the sentiment of tweets as positive or negative (Mozafari et al., 2014);

(4) In the *stage2* data set, workers judge whether a document is related to a topic for document-topic pairs (Tang and Lease, 2011). This dataset was part of the TREC 2011 crowdsourcing track; and

(5) The *rating* data set consists of ratings on a scale of 1 to 10 for products, and the collusive behaviors of workers are identified by obtaining the admission of colluding workers (KhudaBukhsh et al., 2014).

For the first four data sets, all or a partial of the ground true labels are available, and thus we use these data sets to evaluate the performance of ground truth inference of PROCAP. The *rating* data set consists of rating tasks rather than labeling tasks. The ground true labels are not available but the collusive behaviors of workers have been identified, and thus we use this data set for evaluating the performance of collusion detection.

8.2.1 Ground Truth Inference

For the first four datasets, we remove the workers who generated labels for only 1 or 2 tasks, and remove the tasks that receive the labels from only 1 or 2 workers. The summary statistics of the data sets after preprocessing are shown in Table 3. The last row shows the average number of tasks that each pair of workers generate labels to, i.e., the average of $n_{i,j}$. According to the conclusions of the simulation study, the performance of PROCAP generally becomes better with more tasks per worker pair. As can be seen, *bluebird* and *ducks* have relatively large numbers in the last row, and thus we expect higher probability of correctly detecting worker collusion and inferring the ground truth in these two data sets.

	bluebird	*ducks*	*tweets*	*stage2*
Workers	39	53	66	181
Tasks	108	240	1000	3557
Labels	4212	9600	4977	10742
Worker pairs	741	1310	1005	688
Tasks per worker pair	108	142.9	9.9	15.9

Table 3: Summary of the datasets for ground truth inference.

All these data sets consist of binary labeling tasks with $C = 2$, and thus the Class-Dependent DS model is the same as the General DS model. For each data set, we consider three baseline models including the Uniform, Homogeneous, and the General DS models. All available worker-generated labels are used to estimate the ground truth, and the tasks with ground true labels available are used to calculate the accuracy of the inference. In Table 4, we report the accuracy of ground truth inference for the independent-worker baseline models and PROCAP with λ selected by AIC and BIC. We also report the accuracy of majority voting for comparison. If PROCAP has an equal or higher accuracy than the independent-worker baseline model and majority vote, we mark the number as bold in the table. In Table 5, we show the number of detected colluding worker pairs according to BIC for each data set by PROCAP for each baseline model.

For *bluebird* and *stage2*, PROCAP does not show significant improvement over the independent-worker baseline models and majority voting in Table 4. Specifically, the independent-worker General DS model achieves high accuracy in the *bluebird* data set, indicating a good fit to the data. This is further confirmed by Table 5, as no colluding worker is detected in the *bluebird* data set with the General DS model as the baseline. Therefore, PROCAP successfully avoids false detection and overfitting in the *bluebird* data

		bluebird	*ducks*	*tweets*	*stage2*
Majority Voting		0.759	0.692	0.692	0.742
Uniform	Independent	0.741	0.683	0.692	0.830
	Collusion+AIC	**0.759**	**0.746**	**0.692**	**0.830**
	Collusion+BIC	**0.759**	**0.738**	0.689	**0.830**
Homogeneous	Independent	0.583	0.588	0.670	0.830
	Collusion+AIC	0.583	0.575	0.642	**0.830**
	Collusion+BIC	0.583	0.575	0.564	**0.830**
General	Independent	0.880	0.600	0.712	0.723
	Collusion+AIC	**0.880**	0.596	**0.724**	0.645
	Collusion+BIC	**0.880**	0.604	**0.718**	0.645

Table 4: Result of ground truth inference for the four data sets.

	bluebird	*ducks*	*tweets*	*stage2*
Uniform	446	967	494	326
Homogeneous	305	1300	740	227
General	0	301	445	198

Table 5: Detected number of colluding worker pairs for the four data sets.

set with the appropriate baseline model. Although colluding workers are detected in the *stage2* data set, the inference accuracy does not show much difference. One possible reason is that the number of tasks for each worker pair is too small to accurately estimate the parameters. For the *tweets* data set, PROCAP shows a slightly better performance than the baseline models. Since the number of tasks for each worker pair is very small, the estimation may not be accurate, which affects the performance. For the *ducks* data set, PROCAP has a significantly better performance when the Uniform DS model is used as the baseline.

Table 6 reports the average time in seconds until the EM algorithm and the CD algorithm stop for each data set based on a single CPU core, where the numbers in parentheses are the number of iterations. For the four data sets, the EM algorithm (or the CD algorithm) stops when the log-likelihood $\bar{\ell}_p(\boldsymbol{\theta})$ (or the penalized log-likelihood $\bar{f}(\boldsymbol{\theta})$) decreases by a proportion of less than 10^{-8} after the latest iteration. In addition, the EM algorithm is restricted to have less than 50 iterations. As shown in the table, although collusion detection is a complicated problem with huge number of possible colluding combinations, our method greatly decreases the complexity of the problem by considering workers in a pairwise manner, and thus leading to an acceptable computation cost.

In practice, it is possible that additional information is available to further improve the result of ground truth inference. For example, from a prior study or some tasks with known ground truth, we may have a good estimation of the marginal distribution $\boldsymbol{m}$. Thus, in the numerical study, we also implement the methods with the marginal distribution $\boldsymbol{m}$ set to be the value calculated from the ground true labels. The results are shown in Table 7 and Table 8. It turns out that Table 8 is very similar to Table 5 except for the *ducks* data set with the baseline model to be the Homogeneous and General DS models. This is

		bluebird	*ducks*	*tweets*	*stage2*
EM	Uniform	3.6 (45.0)	6.3 (42.6)	0.6 (5.7)	1.2 (14.8)
	Homogeneous	1.6 (16.9)	1.8 (12.7)	1.3 (12.3)	1.4 (17.3)
	General	3.2 (37.5)	5.7 (40.8)	1.9 (19.1)	1.8 (26.5)
CD	Uniform	138.8 (37.5)	218.3 (33.0)	5.3 (7.4)	4.8 (3.8)
	Homogeneous	9.8 (5.7)	13.2 (6.2)	8.1 (5.6)	6.9 (4.7)
	General	28.6 (8.8)	76.0 (12.9)	10.7 (5.6)	10.2 (5.4)

Table 6: Computation cost for the four data sets.

also reflected in Table 7, as the accuracy with these two baseline models for the ducks data set is also greatly different from Table 4. In fact, the performance of PROCAP based on the Homogeneous DS model deteriorates, but the performance with the General DS model greatly improves, indicating a good fit to the data. This shows that we may further improve the performance by incorporating more information.

		bluebird	*ducks*	*tweets*	*stage2*
Majority Voting		0.759	0.692	0.692	0.742
Uniform	Independent	0.759	0.692	0.692	0.817
	Collusion+AIC	**0.759**	**0.725**	**0.692**	**0.830**
	Collusion+BIC	**0.759**	**0.725**	0.690	**0.830**
Homogeneous	Independent	0.713	0.588	0.689	0.807
	Collusion+AIC	0.583	0.413	0.686	0.797
	Collusion+BIC	0.583	0.413	0.688	0.797
General	Independent	0.898	0.613	0.712	0.781
	Collusion+AIC	**0.898**	**0.967**	**0.723**	0.714
	Collusion+BIC	**0.898**	**0.913**	**0.719**	0.728

Table 7: Result of ground truth inference for the four data sets with $\boldsymbol{m}$ set to be the previously estimated value.

	bluebird	*ducks*	*tweets*	*stage2*
Uniform	459	957	494	326
Homogeneous	279	745	487	186
General	0	878	463	194

Table 8: Detected number of colluding worker pairs for the four data sets with $\boldsymbol{m}$ set to be the previously estimated value.

To summarize, we have similar observations as in the simulation study. In particular, if the baseline model is a good fit to the data and the number of tasks for each worker pair is large enough, PROCAP achieves good performance. In addition, incorporating more information may further enhance the performance of PROCAP.

8.2.2 Collusion Detection

The *rating* data set contains the ratings of 20 products from 123 workers with a scale from 1 to 10. The collected ratings are quite dense and each worker rates 19.97 products on average. Although PROCAP is not originally designed for rating tasks, we can still implement the method by transforming the ratings into labels to test the performance in collusion detection. The difference between rating tasks and labeling tasks is that, instead of generating the same label, colluding workers in rating tasks may generate similar but not exactly the same ratings. To address this issue, we transform each pair of ratings for each product into a pair of labels that reflects the colluding possibility. Specifically, let $r_{i,t}$ be the rating of task t given by worker i. Without loss of generality, we define a mapping $\eta(\cdot)$ where $\eta(r) = 1$ if $r \in \{1, 2\}$, $\eta(r) = 2$ if $r \in \{3, 4, 5\}$, $\eta(r) = 3$ if $r \in \{6, 7\}$, and $\eta(r) = 4$ if $r \in \{8, 9, 10\}$. In this way, the mapping $\eta(\cdot)$ transforms a rating in a scale from 1 to 10 into a label with four alternatives. Accordingly, we transform each pair of ratings by

$$(r_{i,t}, r_{j,t}) \to \begin{cases} (\eta(r_{i,t}), \eta(r_{j,t})) \ , & \text{if } |r_{i,t} - r_{j,t}| \geq 2 \ , \\ (\eta(r^{\max}), \eta(r^{\max})) \ , & \text{if } |r_{i,t} - r_{j,t}| \leq 1 \ , \end{cases}$$

where $r^{\max} = \max\{r_{i,t}, r_{j,t}\}$. In other words, if the two original ratings differ by 2 or more, we simply transform them into labels separately according to $\eta(\cdot)$. If the two ratings differ by 1 or less, we consider they may be resulted from collusion, and thus we transform them into the same label, and then PROCAP can determine whether there is indeed a collusion. Additional studies have been conducted with different mappings $\eta(\cdot)$. The results are similar and thus are omitted here.

In Figure 15, we show the true and detected colluding workers based on different baseline models with the tuning parameter λ selected by BIC. The colluding pattern of workers is illustrated as a symmetric matrix, with $H_{i,j}$ to be the ith row and jth column, and the grey scale indicates the value of $H_{i,j}$. A white pixel represents $H_{i,j} = 0$ and a black one represents $H_{i,j} = 1$. The diagonal entries of the matrices are set to be 1, and the workers are permutated such that the true collusion matrix is block-diagonal. By comparing the detected collusion matrices with the true collusion matrix, we can see that with the Class-Dependent or General DS model as the baseline, the collusion pattern becomes quite clear, despite some false detections. In fact, a majority of the false detections can be further eliminated by setting a threshold against the estimated $\hat{H}_{i,j}$. This study shows that PROCAP can be potentially extended to detect collusion in rating tasks as well.

9. Conclusions and Future Work

In crowdsourcing systems, workers are usually not mutually independent but collude with others to gain more rewards. However, existing studies usually assume workers to be independent and thus are vulnerable to worker collusion. This study aims at addressing this issue by detecting the collusive behaviors of workers for labeling tasks in crowdsourcing systems. Based on a baseline model that describes the worker's behavior when working independently of others, we introduce a new parameter $H_{i,j}$ to characterize the probability of collusive behaviors among workers in a pairwise manner, and propose to estimate the parameters by pairwise likelihood estimation. To address the identifiability issue and over-parameterization, we further propose the pairwise profile likelihood estimation, and rely

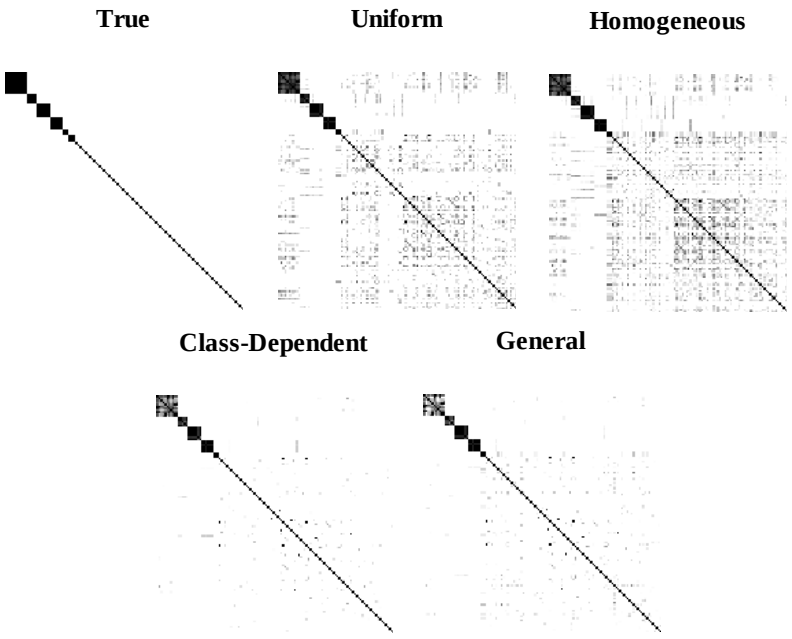

Figure 15: True and detected worker collusion matrices with different baseline models according to BIC for the *rating* data set.

on adaptive LASSO penalty to ensure the sparsity. Then, we investigate the asymptotic properties of the proposed PROCAP method and propose algorithms for deriving numerical solutions. To the best of our knowledge, this is the first statistical model that simultaneously detects worker collusion, learns the capabilities of workers, and infers the ground true labels, with theoretical guarantees. Numerical experiments using simulated data sets and real data sets are conducted to thoroughly test the proposed PROCAP method. We find that PROCAP achieves good performance in ground truth inference and collusion detection when a relatively large amount of data is available and a proper baseline model is adopted.

In the future, there are some important problems worth investigation. First, there are many existing models for ground truth inference in the literature, and the performance of each model heavily depends on the data set. How to select the best model that fits the data set has been an important problem that remains to be solved. This study encounters the problem as well, as the good performance of PROCAP relies on a suitable baseline model. Therefore, one potential future study is to develop a systematic approach for selecting a suitable independent-worker baseline model. Second, both the theoretical results and the numeric experiments show that the performance of PROCAP improves if each worker generates labels for more tasks. However, in practice, the requester may not be able to control the number of labels that each worker generates. Therefore, it is important to explore different manners of incorporating more information into the model to enhance the performance when only limited data is available. For instance, in practice, it is possible that for some tasks, the ground true labels are known, which are usually referred to as gold tasks. These tasks may be incorporated into PROCAP to deliver better results. As

shown in the simulation, one scenario that PROCAP may fail is when a large number of sophisticated workers are involved and always generate the wrong labels. However, such a scenario can be easily detected if some gold tasks are available. Third, this paper focuses on labeling tasks, but similar idea can be extended to other crowdsourcing tasks such as rating tasks, which will be an interesting topic for a future study.

Appendix A. Incorporating the GLAD Model as the Baseline

The GLAD model proposed by Whitehill et al. (2009) assumes the expertise of worker i to be α_i and the difficulty of task t to be $1/\beta_t$, where $\beta_t > 0$. When worker i is independent from other workers, the probability for worker i to generate a correct label for task t is

$$p(L_{i,t} = Y_t) = \sigma(\alpha_i \beta_t) = \frac{1}{1 + \exp(-\alpha_i \beta_t)} \ .$$

Here we use $L_{i,t}$ as the label generated by worker i for task t and Y_t as the ground true label for task t. With the GLAD model as the baseline, the *colluding assumption* changes to

$$p(L_{i,t} = l, L_{j,t} = l' | Z^t_{i,j} = 1, Y_t = c) = I(l = l') p(L_{i,t} = L_{j,t} = l | Z^t_{i,j} = 1, Y_t = c) \ ,$$

where $Z^t_{i,j}$ is the collusion indicator for worker i and worker j on task t. The *non-colluding assumption* changes to

$$p(L_{i,t} = l, L_{j,t} = l' | Z^t_{i,j} = 0, Y_t = c) = p(L_{i,t} = l | Y_t = c) p(L_{j,t} = l' | Y_t = c) \ ,$$

where

$$p(L_{i,t} = l | Y_t = c) = [\sigma(\alpha_i \beta_t)]^{I(l=c)} \left[\frac{1 - \sigma(\alpha_i \beta_t)}{C - 1} \right]^{I(l \neq c)} \ .$$

Then $\bar{\ell}_p(\boldsymbol{\theta})$ can be formulated as in (4) and (5), and $\bar{f}(\boldsymbol{\theta})$ can be derived accordingly.

Appendix B. Derivation of the Pairwise Profile Likelihood

To derive the pairwise profile likelihood, our idea is to find an upper bound of the pairwise log-likelihood with $\boldsymbol{b}_{i,j}$ eliminated. Depending on whether the workers generate the same label or not, the pairwise log-likelihood for the pair of workers (i, j) can be written as

$$\begin{aligned} \ell_p^{(i,j)}(\tilde{\boldsymbol{\theta}}_{i,j}) &= \sum_{l,l' \in \mathcal{C}} n_{i,j}^{l,l'} \log p(L_i = l, L_j = l' | \tilde{\boldsymbol{\theta}}_{i,j}) \\ &= \sum_{l,l' \in \mathcal{C}: l \neq l'} n_{i,j}^{l,l'} \log p(L_i = l, L_j = l' | \tilde{\boldsymbol{\theta}}_{i,j}) + \sum_{l \in \mathcal{C}} n_{i,j}^{l,l} \log p(L_i = L_j = l | \tilde{\boldsymbol{\theta}}_{i,j}) \ . \end{aligned}$$

The first term considers the case when workers i and j generate different labels $l \neq l'$. According to the *colluding assumption*, the two workers must be independent when generating different labels, and thus $\boldsymbol{b}_{i,j}$ is not involved in the first term, i.e.,

$$p(L_i = l, L_j = l' | \tilde{\boldsymbol{\theta}}_{i,j}) = p(L_i = l, L_j = l', Z_{i,j} = 0 | \boldsymbol{\theta}_{i,j}) \ .$$

For the second term, $\boldsymbol{b}_{i,j}$ is related to probabilities $p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j}), l \in \mathcal{C}$, but the overall probability for the workers to generate the same label $p(L_i = L_j|\boldsymbol{\theta}_{i,j}) = \sum_{l\in\mathcal{C}} p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j})$ is independent of $\boldsymbol{b}_{i,j}$, because

$$p(L_i = L_j|\boldsymbol{\theta}_{i,j}) = 1 - p(L_i \neq L_j|\boldsymbol{\theta}_{i,j}) = p(L_i = L_j|Z_{i,j} = 0, \boldsymbol{\theta}_{i,j})(1 - H_{i,j}) + H_{i,j} \ .$$

This motivates us to define the upper bound of the second term using the overall probability $\log p(L_i = L_j|\boldsymbol{\theta}_{i,j})$ according to Jensen's inequality. Specifically, let $e_{i,j} = \sum_{l\in\mathcal{C}} n_{i,j}^{l,l}$, the second term can be rewritten as

$$\sum_{l\in\mathcal{C}} n_{i,j}^{l,l} \log p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j}) = e_{i,j} \sum_{l\in\mathcal{C}} \frac{n_{i,j}^{l,l}}{e_{i,j}} \log \left[\frac{1}{n_{i,j}^{l,l}} p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j}) \right] + \sum_{l} n_{i,j}^{l,l} \log n_{i,j}^{l,l} \ .$$

According to Jensen's inequality, we get

$$\sum_{l\in\mathcal{C}} \frac{n_{i,j}^{l,l}}{e_{i,j}} \log \left[\frac{1}{n_{i,j}^{l,l}} p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j}) \right] \leq \log \left[\frac{1}{e_{i,j}} \sum_{l\in\mathcal{C}} p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j}) \right] \ ,$$

with equality holds if and only if $\frac{1}{n_{i,j}^{l,l}} p(L_i = L_j = l|\tilde{\boldsymbol{\theta}}_{i,j})$ is a constant with respect to l. Therefore, we get

$$\ell_p^{(i,j)}(\tilde{\boldsymbol{\theta}}_{i,j}) \leq \sum_{l,l'\in\mathcal{C}:l\neq l'} n_{i,j}^{l,l'} \log p(L_i = l, L_j = l', Z_{i,j} = 0|\boldsymbol{\theta}_{i,j}) + e_{i,j} \log p(L_i = L_j|\boldsymbol{\theta}_{i,j}) + \sum_{l} n_{i,j}^{l,l} \log n_{i,j}^{l,l} - e_{i,j} \log e_{i,j} \ .$$

By ignoring the constants, we derive the pairwise profile log-likelihood $\bar{\ell}_p(\boldsymbol{\theta})$ with $\boldsymbol{b}_{i,j}$ eliminated.

Appendix C. Proof of Theorems 1-4

Denote $\boldsymbol{\Theta}$ as the domain of $\boldsymbol{\theta}$. It is straightforward to see that $\boldsymbol{\Theta}$ is compact with all entries of $\boldsymbol{\theta}$ between 0 and 1, and $\sum_{l=1}^{C-1} [\boldsymbol{a}_i]_{y,l} \leq 1$, $\sum_{y=1}^{C-1} m_y \leq 1$. We derive the asymptotic properties following the literature (Fan and Li, 2001; Wang and Leng, 2007) with a special treatment of the case when the true $H_{i,j}^* = 0$.

C.1 Proof of Theorem 1

Consider the pairwise profile log-likelihood function $\bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ for the pair of workers (i, j). Let $\boldsymbol{\theta}_{i,j}^*$ be the true parameter. Since

$$\frac{n_{i,j}^{l,l'}}{n_{i,j}} \to p(L_i = l, L_j = l', Z_{i,j} = 0|\boldsymbol{\theta}_{i,j}^*) \ ,$$

$$\frac{\sum_{l\in\mathcal{C}} n_{i,j}^{l,l}}{n_{i,j}} \to p(L_i = L_j|\boldsymbol{\theta}_{i,j}^*) \ ,$$

as $n_{i,j} \to \infty$, it is straightforward to see the pointwise convergence

$$\frac{1}{n_{i,j}} \bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) \to \bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) \ .$$

Here

$$\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) = \sum_{l,l' \in \mathcal{C}: l \neq l'} p(L_i = l, L_j = l', Z_{i,j} = 0 | \boldsymbol{\theta}^*_{i,j}) \log p(L_i = l, L_j = l', Z_{i,j} = 0 | \boldsymbol{\theta}_{i,j}) \\ + p(L_i = L_j | \boldsymbol{\theta}^*_{i,j}) \log p(L_i = L_j | \boldsymbol{\theta}_{i,j}) \ ,$$

where

$$p(L_i = l, L_j = l', Z_{i,j} = 0 | \boldsymbol{\theta}_{i,j}) = p(L_i = l, L_j = l' | Z_{i,j} = 0, \boldsymbol{\theta}_{i,j})(1 - H_{i,j}) \ ,$$

and

$$p(L_i = L_j | \boldsymbol{\theta}_{i,j}) = H_{i,j} + (1 - H_{i,j}) p(L_i = L_j | Z_{i,j} = 0, \boldsymbol{\theta}_{i,j}) \ .$$

If we regard the pairwise profile likelihood as an M-estimator, we can verify that $\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ is the expectation of the estimation function, i.e.,

$$\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) = E_{\boldsymbol{\theta}^*_{i,j}}[\mathcal{M}_{i,j}(l_{i,t}, l_{j,t}; \boldsymbol{\theta}_{i,j})] \ .$$

In fact, it is straightforward to verify that the convergence is uniform.

In addition, according to Jensen's inequality,

$$\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j}) - \bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}^*_{i,j}) \leq 0 \ . \tag{10}$$

Thus $\boldsymbol{\theta}^*_{i,j}$ is a maximizer of $\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$. Therefore, based on Theorem 5.7 of van der Vaart (2000) and extending the result to the overall pairwise profile log-likelihood $\bar{\ell}_p(\boldsymbol{\theta})$, we conclude that $\hat{\boldsymbol{\theta}} \to \boldsymbol{\theta}^*$ in probability.

It is worth noting that for a pair of workers (i, j), the implicit constraint on transitive $z^t_{i,j}$ does not affect the likelihood function $\bar{\ell}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ and $\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$, because the constraint only applies to three or more workers. As discussed above, $\hat{\boldsymbol{\theta}}_{i,j}$ converges to $\boldsymbol{\theta}^*_{i,j}$ even without the constraint. Therefore, the constraint is not needed for parameter estimation.

To show the convergence rate, denote $\boldsymbol{\Theta}_{i,j}$ as the domain of $\boldsymbol{\theta}_{i,j}$. Under the regularity condition, the true parameters $[\boldsymbol{a}^*_i]_{y,l} \in (0, 1)$, $m^*_y \in (0, 1)$, and $H^*_{i,j} \in [0, 1)$. If $H^*_{i,j} > 0$, then $\boldsymbol{\theta}^*_{i,j}$ is an inner point of $\boldsymbol{\Theta}_{i,j}$ and thus $\boldsymbol{\theta}^*_{i,j}$ is a local maximizer of $\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ with a zero gradient and a negative definite Hessian matrix. If $H^*_{i,j} = 0$, then $\boldsymbol{\theta}^*_{i,j}$ is on the boundary of $\boldsymbol{\Theta}_{i,j}$. In this case, it is straightforward to verify that (10) still holds if $H_{i,j}$ is a small negative number, and thus $\bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$ still has a zero gradient and a negative definite Hessian matrix at $\boldsymbol{\theta}^*_{i,j}$. As a result, the root-$n$ convergence rate follows Corollary 5.53 of van der Vaart (2000).

C.2 Proof of Theorem 2

The proof generally follows Fan and Li (2001), but has been tailored for our model with the consideration of the case when $H^*_{i,j} = 0$. To prove Theorem 2, we need to show that for an arbitrarily small $\epsilon > 0$, there exists a sufficiently large constant $\mathcal{D}$ such that

$$p\left[\sup_{\boldsymbol{u}} \bar{f}\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) < \bar{f}(\boldsymbol{\theta}^*)\right] \geq 1 - \epsilon \ ,$$

where $\|\boldsymbol{u}\| = \mathcal{D}$ and $\boldsymbol{\theta}^* + \boldsymbol{u}/\sqrt{T} \in \boldsymbol{\Theta}$. This means that there is a local maximizer $\hat{\boldsymbol{\theta}}$ in the space $\{\boldsymbol{\theta}^* + \boldsymbol{u}/\sqrt{T} \in \boldsymbol{\Theta} : \|\boldsymbol{u}\| \leq \mathcal{D}\}$ with a probability of at least $1 - \epsilon$, and thus $\sqrt{T}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) = O_p(1)$. We consider

$$\bar{f}\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) - \bar{f}(\boldsymbol{\theta}^*) = \bar{\ell}_p\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) - \bar{\ell}_p(\boldsymbol{\theta}^*) - \lambda_T\sqrt{T}\sum_{(i,j)\in\mathcal{P}} w_{i,j}u_{i,j} \ ,$$

where $u_{i,j}$ is the entry of $\boldsymbol{u}$ corresponding to $H_{i,j}$. For any $(i,j) \in \mathcal{P}_0$, since $H^*_{i,j} = 0$ and $H^*_{i,j} + u_{i,j}/\sqrt{T} \geq 0$, we must have $u_{i,j} \geq 0$, and thus

$$\bar{f}\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) - \bar{f}(\boldsymbol{\theta}^*) \leq \bar{\ell}_p\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) - \bar{\ell}_p(\boldsymbol{\theta}^*) - \lambda_T\sqrt{T}\sum_{(i,j)\in\mathcal{P}_1} w_{i,j}u_{i,j} \ .$$

According to Taylor expansion,

$$\bar{\ell}_p\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) = \bar{\ell}_p\left(\boldsymbol{\theta}^*\right) + \frac{1}{\sqrt{T}}\boldsymbol{u}^T\bar{\ell}'_p(\boldsymbol{\theta}^*) + \frac{1}{2}\boldsymbol{u}^T\bar{\mathcal{L}}''_p(\boldsymbol{\theta}^*_{i,j})\boldsymbol{u}\{1 + o_p(1)\} \ ,$$

where $\bar{\ell}'_p(\boldsymbol{\theta}^*)$ is the gradient of $\bar{\ell}_p(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$, $\bar{\mathcal{L}}''_p(\boldsymbol{\theta}^*)$ is the Hessian matrix of $\bar{\mathcal{L}}_p(\boldsymbol{\theta})$ at $\boldsymbol{\theta}^*$, and $\bar{\mathcal{L}}_p(\boldsymbol{\theta}) = \sum_{(i,j)\in\mathcal{P}} \bar{\mathcal{L}}_p^{(i,j)}(\boldsymbol{\theta}_{i,j})$. Therefore,

$$\bar{f}\left(\boldsymbol{\theta}^* + \frac{\boldsymbol{u}}{\sqrt{T}}\right) - \bar{f}(\boldsymbol{\theta}^*) \leq \frac{1}{\sqrt{T}}\boldsymbol{u}^T\bar{\ell}'_p(\boldsymbol{\theta}^*) + \frac{1}{2}\boldsymbol{u}^T\bar{\mathcal{L}}''_p(\boldsymbol{\theta}^*_{i,j})\boldsymbol{u}\{1 + o_p(1)\} - \lambda_T\sqrt{T}\sum_{(i,j)\in\mathcal{P}_1} w_{i,j}u_{i,j} \ .$$

From the proof the Theorem 1, we know that no matter whether $\boldsymbol{\theta}^*$ is an inner point of $\boldsymbol{\theta}$ or on the boundary with some $H^*_{i,j} = 0$, $\bar{\mathcal{L}}''_p(\boldsymbol{\theta}^*_{i,j})$ is always negative definite and $\bar{\ell}'_p(\boldsymbol{\theta}^*)/\sqrt{T} = O_p(1)$. Thus, the first term of the right-hand side of the above equation is dominated by the second term with a sufficiently large $\|\boldsymbol{u}\|$. In addition, the third term is bounded by $|\mathcal{P}_1|\sqrt{T}g_T^1\|\boldsymbol{u}\|$, where $|\mathcal{P}_1|$ is the number of elements of $\mathcal{P}_1$, and thus with the condition $\sqrt{T}g_T^1 \to 0$, the third term is also dominated by the second term. Therefore, $\bar{f}\left(\boldsymbol{\theta}^* + \boldsymbol{u}/\sqrt{T}\right) - \bar{f}(\boldsymbol{\theta}^*) \leq 0$ by selecting a sufficiently large $\mathcal{D}$. This completes the proof.

C.3 Proof of Theorem 3

To show part (a), let $\hat{\boldsymbol{\theta}}$ be a root-n consistent estimator of $\boldsymbol{\theta}^*$ from Theorem 2. For any $(i,j) \in \mathcal{P}_0$, our idea is to prove that if $\hat{H}_{i,j} > 0$, then $\partial\bar{f}(\hat{\boldsymbol{\theta}})/\partial H_{i,j} < 0$, which contradicts with our assumption that $\hat{\boldsymbol{\theta}}$ is a local maximizer as we can further increase $\bar{f}(\hat{\boldsymbol{\theta}})$ by decreasing $\hat{H}_{i,j}$. Specifically, we consider

$$\frac{1}{\sqrt{T}}\frac{\partial\bar{f}(\hat{\boldsymbol{\theta}})}{\partial H_{i,j}} = \frac{1}{\sqrt{T}}\gamma(\hat{\boldsymbol{\theta}}) - \sqrt{T}\lambda_T w_{i,j} \ ,$$

where

$$\gamma(\boldsymbol{\theta}) = \frac{\partial \bar{\ell}_p(\boldsymbol{\theta})}{\partial H_{i,j}} .$$

According to mean value theorem,

$$\frac{1}{\sqrt{T}}\gamma(\hat{\boldsymbol{\theta}}) = \frac{1}{\sqrt{T}}\gamma(\boldsymbol{\theta}^*) + (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)^T \frac{1}{\sqrt{T}}\gamma'\left(\alpha\boldsymbol{\theta}^* + (1-\alpha)\hat{\boldsymbol{\theta}}\right) ,$$

where $0 < \alpha < 1$. The first term of the right-hand side of the equation is $O_p(1)$. Since $\sqrt{T}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) = O_p(1)$, and

$$\frac{1}{T}\frac{\partial^2 \bar{\ell}_p(\boldsymbol{\theta}^*)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^T} = \bar{\mathcal{L}}_p''(\boldsymbol{\theta}^*) + o_p(1) ,$$

the second term is also $O_p(1)$. Therefore, $\gamma(\hat{\boldsymbol{\theta}})/\sqrt{T}$ is $O_p(1)$. With the condition that $\sqrt{T}g_T^0 \to \infty$, for any $(i,j) \in \mathcal{P}_0$, $\sqrt{T}\lambda_T w_{i,j} \geq \sqrt{T}g_T^0 \to \infty$, and thus $\frac{1}{\sqrt{T}}\frac{\partial \bar{f}(\hat{\boldsymbol{\theta}})}{\partial H_{i,j}} < 0$. This completes the proof for part (a).

For part (b), consider any $(i,j) \in \mathcal{P}_1$, we know

$$\frac{1}{\sqrt{T}}\frac{\partial \bar{f}(\hat{\boldsymbol{\theta}})}{\partial H_{i,j}} = \frac{1}{\sqrt{T}}\gamma(\hat{\boldsymbol{\theta}}) - \sqrt{T}\lambda_T w_{i,j} = 0 .$$

Since $\sqrt{T}\lambda_T w_{i,j} \leq \sqrt{T}g_T^1 \to 0$, following similar argument with part (a), we get

$$\frac{1}{\sqrt{T}}\gamma(\hat{\boldsymbol{\theta}}) = \frac{1}{\sqrt{T}}\gamma(\boldsymbol{\theta}^*) + (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)^T \frac{1}{\sqrt{T}}\gamma'\left(\alpha\boldsymbol{\theta}^* + (1-\alpha)\hat{\boldsymbol{\theta}}\right) \to 0 .$$

According to part (a), with probability tending to 1, $\hat{\boldsymbol{\theta}}_0 = \mathbf{0} = \boldsymbol{\theta}_0^*$, and thus $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^* = (\mathbf{0}, \hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c^*)$. Then, we get

$$\frac{1}{\sqrt{T}}\gamma(\boldsymbol{\theta}^*) + (\hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c^*)^T \frac{1}{\sqrt{T}}\frac{\partial\gamma\left(\alpha\boldsymbol{\theta}^* + (1-\alpha)\hat{\boldsymbol{\theta}}\right)}{\partial\boldsymbol{\theta}_c} \to 0 .$$

Following the same procedure for each entry of $\boldsymbol{\theta}_c$ and combining the result into a matrix form, we get

$$\frac{1}{\sqrt{T}}\frac{\partial \bar{\ell}_p(\boldsymbol{\theta}^*)}{\partial\boldsymbol{\theta}_c} + (\hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c^*)^T \frac{1}{\sqrt{T}}\frac{\partial^2 \bar{\ell}_p\left(\alpha\boldsymbol{\theta}^* + (1-\alpha)\hat{\boldsymbol{\theta}}\right)}{\partial\boldsymbol{\theta}_c\partial\boldsymbol{\theta}_c^T} \to \mathbf{0} .$$

Since

$$\frac{1}{\sqrt{T}}\frac{\partial \bar{\ell}_p(\boldsymbol{\theta}^*)}{\partial\boldsymbol{\theta}_c} \to N(\mathbf{0}, \boldsymbol{\Sigma}_c) ,$$
$$\frac{1}{T}\frac{\partial^2 \bar{\ell}_p\left(\alpha\boldsymbol{\theta}^* + (1-\alpha)\hat{\boldsymbol{\theta}}\right)}{\partial\boldsymbol{\theta}_c\partial\boldsymbol{\theta}_c^T} \to \boldsymbol{\Sigma}_c ,$$

where

$$\boldsymbol{\Sigma}_c = \frac{\partial^2 \mathcal{L}_p(\boldsymbol{\theta}^*)}{\partial\boldsymbol{\theta}_c\boldsymbol{\theta}_c^T} ,$$

we can easily get $\sqrt{T}(\hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c^*) \to N(\mathbf{0}, \boldsymbol{\Sigma}_c)$.

C.4 Proof of Theorem 4

We consider the underfitted case and overfitted case separately.

Case 1: If $\lambda \in \mathcal{R}_+^u$, the corresponding model is underfitted. We consider

$$\frac{1}{T}\left(\mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\lambda)} - \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)}\right) = \frac{2}{T}\{\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\lambda)]\} + \frac{\log T}{T}\left(\left|\hat{\mathcal{P}}_1(\lambda)\right| - \left|\hat{\mathcal{P}}_1(\tilde{\lambda}_T)\right|\right) ,$$

where $\left|\hat{\mathcal{P}}_1(\lambda)\right| = \sum_{(i,j)\in\mathcal{P}} I\left(\hat{H}_{i,j}(\lambda) > 0\right)$ is the number of colluding worker pairs indicated by $\hat{\boldsymbol{\theta}}(\lambda)$. Obviously, the second term $\frac{\log T}{T}\left(\left|\hat{\mathcal{P}}_1(\lambda)\right| - \left|\hat{\mathcal{P}}_1(\tilde{\lambda}_T)\right|\right) \to 0$, and thus we focus on the first term. Given a set of independent worker pairs $\mathcal{P}_0$, we can obtain $\hat{\boldsymbol{\theta}}(\mathcal{P}_0)$ as the maximizer of $\bar{\ell}_p(\boldsymbol{\theta})$ with constraints $H_{i,j} = 0$ for any $(i,j) \in \mathcal{P}_0$. Thus $\hat{\boldsymbol{\theta}}\left(\hat{\mathcal{P}}_0(\lambda)\right)$ is the estimation based on the model indicated by $\hat{\boldsymbol{\theta}}(\lambda)$. We know

$$\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\lambda)] \le \bar{\ell}_p\left[\hat{\boldsymbol{\theta}}\left(\hat{\mathcal{P}}_0(\lambda)\right)\right] .$$

In addition, within all underfitted models with $\mathcal{P}_0 \not\subset \mathcal{P}_0^*$, we can find the one with the maximum $\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0)]$, denoted as $\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})]$. We get

$$\bar{\ell}_p\left[\hat{\boldsymbol{\theta}}\left(\hat{\mathcal{P}}_0(\lambda)\right)\right] \le \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})] .$$

Therefore

$$\frac{2}{T}\{\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\lambda)]\} \ge \frac{2}{T}\{\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})]\} .$$

Since $\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)$ is consistent, from the proof of Theorem 1, we know

$$\frac{1}{T}\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] \to \bar{\mathcal{L}}_p(\boldsymbol{\theta}^*) ,$$
$$\frac{1}{T}\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})] \to \bar{\mathcal{L}}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})] .$$

We have shown that $\boldsymbol{\theta}^*$ is the maximizer of $\bar{\mathcal{L}}_p(\boldsymbol{\theta})$, thus $\bar{\mathcal{L}}_p(\boldsymbol{\theta}^*) > \bar{\mathcal{L}}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})]$. Therefore, with probability tending to 1, we get $\frac{2}{T}\{\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})]\} > 0$, which means $\mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\lambda)} > \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)}$. Consequently,

$$p\left(\inf_{\lambda\in\mathcal{R}_+^u} \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\lambda)} - \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)}\right) \to 1 .$$

Case 2: If $\lambda \in \mathcal{R}_+^o$, the corresponding model is overfitted. We consider

$$\mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\lambda)} - \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)} = 2\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - 2\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\lambda)] + \log T\left(\left|\hat{\mathcal{P}}_1(\lambda)\right| - \left|\hat{\mathcal{P}}_1(\tilde{\lambda}_T)\right|\right) .$$

Since $\tilde{\lambda}_T$ satisfies the conditions of Theorem 2 and Theorem 3, with probability tending to 1, the corresponding model is correct, i.e., $\hat{\mathcal{P}}_1(\tilde{\lambda}_T) = \mathcal{P}_1^*$. Since $\hat{\mathcal{P}}_1(\lambda)$ corresponds to an overfitted model, we get $\left|\hat{\mathcal{P}}_1(\lambda)\right| - |\mathcal{P}_1^*| \ge 1$, thus

$$\log T\left(\left|\hat{\mathcal{P}}_1(\lambda)\right| - \left|\hat{\mathcal{P}}_1(\tilde{\lambda}_T)\right|\right) \ge \log T .$$

In addition, following similar argument with *Case 1*, we get

$$\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\lambda)] \geq \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p\left[\hat{\boldsymbol{\theta}}\left(\hat{\mathcal{P}}_0(\lambda)\right)\right] \geq \bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p\left[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})\right] \ ,$$

where $\mathcal{P}_0^{\max}$ is the model within all overfitted models $\mathcal{P}_0 \subset \mathcal{P}_0^*$, $\mathcal{P}_0 \neq \mathcal{P}_0^*$ that has the maximum $\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\mathcal{P}_0)]$. Since both $\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)$ and $\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})$ are root-n consistent, $\bar{\ell}_p[\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)] - \bar{\ell}_p\left[\hat{\boldsymbol{\theta}}(\mathcal{P}_0^{\max})\right]$ is $O_p(1)$, and thus $\mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\lambda)} - \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)}$ is dominated by $\log T$. Therefore,

$$p\left(\inf_{\lambda \in \mathcal{R}_+^o} \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\lambda)} - \mathrm{BIC}_{\hat{\boldsymbol{\theta}}(\tilde{\lambda}_T)}\right) \to 1 \ .$$

By combining the two cases, we finish the proof of Theorem 4.

Appendix D. Proof of Theorem 5

Consider

$$\begin{aligned}
\mathcal{G}(\boldsymbol{\theta}|\boldsymbol{\theta}^k) &= \bar{\ell}_p\left(\boldsymbol{\theta}\right) - \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k) \\
&= -\sum_{(i,j)\in\mathcal{P}} \Bigg[\sum_{l,l'\in\mathcal{C}:l\neq l'} n_{i,j}^{l,l'} \sum_{y\in\mathcal{C}} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) \log \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}) \\
&\qquad + \left(\sum_{l\in\mathcal{C}} n_{i,j}^{l,l}\right) \sum_{y,l\in\mathcal{C}} \sum_{z\in\{0,1\}} \mathcal{Q}_{i,j}^2(y,z,l|\boldsymbol{\theta}_{i,j}^k) \log \mathcal{Q}_{i,j}^2(y,z,l|\boldsymbol{\theta}_{i,j}) \Bigg] \ .
\end{aligned}$$

According to Jensen's inequality, it is straightforward to show that $\mathcal{G}(\boldsymbol{\theta}^k|\boldsymbol{\theta}^k) \leq \mathcal{G}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$. Therefore, if $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k) \geq \mathcal{Q}(\boldsymbol{\theta}^k|\boldsymbol{\theta}^k)$, then $\bar{\ell}_p\left(\boldsymbol{\theta}\right) \geq \bar{\ell}_p\left(\boldsymbol{\theta}^k\right)$.

Appendix E. Derivation of the Updating Equations of Section 5.1

We know

$$\log p(L_i = l, L_j = l', Y = y, Z_{i,j} = 0|\boldsymbol{\theta}_{i,j}) = \log(1 - H_{i,j}) + \log m_y + \log[\boldsymbol{a}_i]_{y,l} + \log[\boldsymbol{a}_j]_{y,l'} \ ,$$

$$\begin{aligned}
&\log p(L_i = L_j = l, Y = y, Z_{i,j} = 1|\boldsymbol{\theta}_{i,j}) \\
&\qquad = \log H_{i,j} + \log m_y + \log p(L_i = L_j = l|Y = y, Z_{i,j} = 1, \boldsymbol{\theta}_{i,j}) \ .
\end{aligned}$$

To maximize $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k)$ under the constraints $\sum_{y\in\mathcal{C}} m_y = 1$ and $\sum_{l\in\mathcal{C}}[\boldsymbol{a}_i]_{y,l} = 1$, we consider Lagrangian multipliers and maximize $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^k) - \lambda_m(\sum_{y\in\mathcal{C}} m_y - 1) - \lambda_\alpha^i(\sum_{l\in\mathcal{C}}[\boldsymbol{a}_i]_{y,l} - 1)$. For m_y, by taking the first derivative, we obtain

$$\begin{aligned}
&\frac{1}{m_y} \sum_{(i,j)\in\mathcal{P}} \Bigg[\sum_{l,l'\in\mathcal{C}:l\neq l'} n_{i,j}^{l,l'} \mathcal{Q}_{i,j}^1(y|0,l,l',\boldsymbol{\theta}_{i,j}^k) \\
&\qquad + \left(\sum_{l\in\mathcal{C}} n_{i,j}^{l,l}\right) \left(\sum_{l\in\mathcal{C}} \mathcal{Q}_{i,j}^2(y,0,l|\boldsymbol{\theta}_{i,j}^k) + \sum_{l\in\mathcal{C}} \mathcal{Q}_{i,j}^2(y,1,l|\boldsymbol{\theta}_{i,j}^k)\right)\Bigg] - \lambda_m = 0 \ .
\end{aligned}$$

Since $p(Y = y|L_i = L_j, \boldsymbol{\theta}^k_{i,j}) = \sum_{l\in\mathcal{C}} \mathcal{Q}^2_{i,j}(y, 0, l|\boldsymbol{\theta}^k_{i,j}) + \sum_{l\in\mathcal{C}} \mathcal{Q}^2_{i,j}(y, 1, l|\boldsymbol{\theta}^k_{i,j})$, we obtain the updating equation for m_y. Similarly, for $[\boldsymbol{a}_i]_{y,l}$, the first derivative is

$$\frac{1}{[\boldsymbol{a}_i]_{y,l}} \sum_{j\in\mathcal{W}:j\neq i} \left[\sum_{l'\in\mathcal{C}:l'\neq l} n^{l,l'}_{i,j} \mathcal{Q}^1_{i,j}(y|0, l, l', \boldsymbol{\theta}^k_{i,j}) + \left(\sum_{l'\in\mathcal{C}} n^{l',l'}_{i,j} \right) \mathcal{Q}^2_{i,j}(y, 0, l|\boldsymbol{\theta}^k_{i,j}) \right] - \lambda^i_\alpha = 0 \ .$$

Then we obtain the updating equation for $[\boldsymbol{a}_i]_{y,l}$. At last, the first derivative for $H_{i,j}$ is

$$\begin{aligned} &- \frac{1}{1 - H_{i,j}} \sum_{l,l'\in\mathcal{C}:l\neq l'} n^{l,l'}_{i,j} \sum_{y\in\mathcal{C}} \mathcal{Q}^1_{i,j}(y|0, l, l', \boldsymbol{\theta}^k_{i,j}) \\ &\qquad + \left(\sum_{l\in\mathcal{C}} n^{l,l}_{i,j} \right) \left[-\frac{1}{1 - H_{i,j}} \sum_{y,l} \mathcal{Q}^2_{i,j}(y, 0, l|\boldsymbol{\theta}^k_{i,j}) + \frac{1}{H_{i,j}} \sum_{y,l} \mathcal{Q}^2_{i,j}(y, 1, l|\boldsymbol{\theta}^k_{i,j}) \right] = 0 \ , \end{aligned}$$

leading to the updating equation for $H_{i,j}$.

References

Mohammad Allahbakhsh, Aleksandar Ignjatovic, Boualem Benatallah, Elisa Bertino, and Norman Foo. Collusion detection in online rating systems. In *Asia-Pacific Web Conference*, pages 196–207. Springer, 2013.

Wei Bi, Liwei Wang, James T. Kwok, and Zhuowen Tu. Learning to predict from crowdsourced data. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 82–91, 2014.

Peng-Peng Chen, Hai-Long Sun, Yi-Li Fang, and Jin-Peng Huai. Collusion-proof result inference in crowdsourcing. *Journal of Computer Science and Technology*, 33(2):351–365, 2018.

Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit players. Predicting protein structures with a multiplayer online game. *Nature*, 466(7307):756–760, 2010.

Alexander P. Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Chrysanthos Dellarocas. Strategic manipulation of internet opinion forums: Implications for consumers and firms. *Management Science*, 52(10):1577–1593, 2006.

Pinar Donmez, Jaime Carbonell, and Jeff Schneider. A probabilistic framework to learn from multiple annotators with time-varying accuracy. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 826–837. SIAM, 2010.

John R. Douceur. The sybil attack. In *International Workshop on Peer-to-Peer Systems*, pages 251–260. Springer, 2002.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Xin Gao and Peter X-K Song. Composite likelihood EM algorithm with applications to multivariate hidden Markov model. *Statistica Sinica*, pages 165–185, 2011.

Arpita Ghosh, Satyen Kale, and Preston McAfee. Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 167–176, 2011.

Mary L. Gray, Siddharth Suri, Syed Shoaib Ali, and Deepti Kulkarni. The crowd is a collaborative network. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 134–147, 2016.

Amy Harmon. Amazon glitch unmasks war of reviewers. *The New York Times*, 2004. URL `https://www.nytimes.com/2004/02/14/us/amazon-glitch-unmasks-war-of-reviewers.html`.

Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Identifying unreliable and adversarial workers in crowdsourced labeling tasks. *Journal of Machine Learning Research*, 18(1):1–67, 2017.

Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. A convex formulation for learning from crowds. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 73–79, 2012.

Hiroshi Kajino, Yuta Tsuboi, and Hisashi Kashima. Clustering crowds. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, pages 1120–1127, 2013.

Ece Kamar, Ashish Kapoor, and Eric Horvitz. Identifying and accounting for task-dependent bias in crowdsourcing. In *Proceedings of the Third AAAI Conference on Human Computation and Crowdsourcing*, pages 92–101, 2015.

Yiannis Kamarianakis, Wei Shen, and Laura Wynter. Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO. *Applied Stochastic Models in Business and Industry*, 28(4):297–315, 2012.

David R. Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. In *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pages 81–92, 2013.

Ashiqur R. KhudaBukhsh, Jaime G. Carbonell, and Peter J. Jansen. Detecting non-adversarial collusion in crowdsourcing. In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing*, pages 104–111, 2014.

Minhee Kim, Changyue Song, and Kaibo Liu. A generic health index approach for multisensor degradation modeling and sensor selection. *IEEE Transactions on Automation Science and Engineering*, 16(3):1426–1437, 2019.

Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*, 2014.

Chris Lintott, Kevin Schawinski, Steven Bamford, Anže Slosar, Kate Land, Daniel Thomas, Edd Edmondson, Karen Masters, Robert C. Nichol, M. Jordan Raddick, et al. Galaxy zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410(1):166–178, 2011.

Qiang Liu, Jian Peng, and Alexander T. Ihler. Variational inference for crowdsourcing. *Advances in Neural Information Processing Systems*, 25:692–700, 2012.

Yuhong Liu, Yafei Yang, and Yan Lindsay Sun. Detection of collusion behaviors in online reputation systems. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pages 1368–1372. IEEE, 2008.

Yuli Liu, Yiqun Liu, Ke Zhou, Min Zhang, and Shaoping Ma. Detecting collusive spamming activities in community question answering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1073–1082, 2017.

Pablo G. Moreno, Antonio Artés-Rodríguez, Yee W. Teh, and Fernando Perez-Cruz. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 16:1607–1627, 2015.

Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, and Samuel Madden. Scaling up crowd-sourcing to very large datasets: a case for active learning. *Proceedings of the VLDB Endowment*, 8(2):125–136, 2014.

Vikas C. Raykar and Shipeng Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13:491–518, 2012.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo H. Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

Mathieu Ribatet, Daniel Cooley, and Anthony C. Davison. Bayesian inference from composite likelihoods, with an application to spatial extremes. *Statistica Sinica*, 22:813–845, 2012.

Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y. Ng. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, 2008.

Changyue Song, Kaibo Liu, and Xi Zhang. A generic framework for multisensor degradation modeling based on supervised classification and failure surface. *IISE Transactions*, 51(11): 1288–1302, 2019.

Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR 2011 Workshop on Crowdsourcing for Information Retrieval (CIR)*, pages 1–6, 2011.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

A. W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 2000.

Cristiano Varin, Nancy Reid, and David Firth. An overview of composite likelihood methods. *Statistica Sinica*, 21(1):5–42, 2011.

Jennifer Wortman Vaughan. Making better use of the crowd: How crowdsourcing can advance machine learning research. *Journal of Machine Learning Research*, 18:1–46, 2018.

Matteo Venanzi, John Guiver, Gabriella Kazai, Pushmeet Kohli, and Milad Shokouhi. Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 155–164, 2014.

Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013.

Hansheng Wang and Chenlei Leng. Unified LASSO estimation by least squares approximation. *Journal of the American Statistical Association*, 102(479):1039–1048, 2007.

Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. The multidimensional wisdom of crowds. *Advances in Neural Information Processing Systems*, 23:2424–2432, 2010.

Jacob Whitehill, Paul Ruvolo, Ting-fan Wu, Jacob Bergsma, and Javier Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems*, 22:2035–2043, 2009.

Aifang Xu, Xiaonan Feng, and Ye Tian. Revealing, characterizing, and detecting crowdsourcing spammers: A case study in community Q&A. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 2533–2541. IEEE, 2015.

Chang Xu. Detecting collusive spammers in online review communities. In *Proceedings of the Sixth Workshop on Ph. D. Students in Information and Knowledge Management*, pages 33–40, 2013.

Yan Yan, Rómer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 932–939, 2010.

Ming Yin, Mary L. Gray, Siddharth Suri, and Jennifer Wortman Vaughan. The communication network within the crowd. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1293–1303, 2016.

Jing Zhang, Xindong Wu, and Victor S Sheng. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review*, 46(4):543–576, 2016.

Changliang Zou, Wei Jiang, and Fugee Tsung. A LASSO-based diagnostic framework for multivariate statistical process control. *Technometrics*, 53(3):297–309, 2011.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.