Journal of Machine Learning Research 18 (2017) 1-67 Submitted 12/15; Revised 12/16; Published 9/17

Identifying Unreliable and Adversarial Workers in Crowdsourced Labeling Tasks

Srikanth Jagabathula SJAGABAT@STERN.NYU.EDU
Department of Information, Operations, and Management Sciences
Leonard N. Stern School of Business
44 West Fourth Street
New York University, NY 10012, USA

Lakshminarayanan Subramanian LAKSHMI@CS.NYU.EDU
Ashwin Venkataraman ASHWIN@CS.NYU.EDU
Department of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York University, NY 10012, USA

Editor: Le Song

Abstract

We study the problem of identifying *unreliable* and *adversarial* workers in crowdsourcing systems where workers (or users) provide labels for tasks (or items). Most existing studies assume that worker responses follow specific probabilistic models; however, recent evidence shows the presence of workers adopting non-random or even malicious strategies. To account for such workers, we suppose that workers comprise a mixture of honest and adversarial workers. *Honest* workers may be reliable or unreliable, and they provide labels according to an unknown but explicit probabilistic model. *Adversaries* adopt labeling strategies different from those of honest workers, whether probabilistic or not. We propose two reputation algorithms to identify unreliable honest workers and adversarial workers from *only* their responses. Our algorithms assume that honest workers are in the majority, and they classify workers with outlier label patterns as adversaries. Theoretically, we show that our algorithms successfully identify unreliable honest workers, workers adopting deterministic strategies, and worst-case *sophisticated* adversaries who can adopt arbitrary labeling strategies to degrade the accuracy of the inferred task labels. Empirically, we show that filtering out outliers using our algorithms can significantly improve the accuracy of several state-of-the-art label aggregation algorithms in real-world crowdsourcing datasets.

Keywords: crowdsourcing, reputation, adversary, outliers

1. Introduction

The growing popularity of online crowdsourcing services like Amazon Mechanical Turk and CrowdFlower has made it easy to collect low-cost labels from the crowd to generate training datasets for machine learning applications. Unfortunately, these labels are typically of low quality because of unintentional or intentional inaccuracies introduced by unreliable and malicious workers (Kittur et al., 2008; Le et al., 2010). Determining correct labels of tasks from such noisy labels is challenging because the reliabilities or qualities of workers are

often unknown. While one may use "gold standard" tasks—whose true label is already known—to identify low-reliability workers (Snow et al., 2008; Downs et al., 2010; Le et al., 2010), accessing true labels for a sufficient number of tasks can be difficult and expensive. To address these challenges, a common solution is to use *redundancy* (Sheng et al., 2008), that is, collecting multiple labels for each task and assigning multiple tasks to each worker. Given the redundant labels, most existing studies make specific probabilistic assumptions on how individual workers provide labels and propose techniques to infer true task labels given workers' responses. Common probabilistic models include the one-coin model (Zhang et al., 2014), two-coin model (Raykar and Yu, 2012), and the general Dawid-Skene model (Dawid and Skene, 1979). For example, the "one-coin" model assumes that each worker w provides the correct label to an assigned task with probability p_w and (an) incorrect label with probability $1 - p_w$. The parameter p_w thus measures the reliability of worker w.

While most existing work relies on explicit probabilistic models, recent studies (Vuurens et al., 2011; Difallah et al., 2012) and anecdotal evidence show that worker labeling strategies may not be probabilistic in practice. For instance, for binary classification tasks, workers may adopt strategies that: (a) uniformly label all tasks $+1$ if it is known that $+1$ labels are more prevalent than -1 among true labels in the corpus of tasks; [1] (b) provide accurate labels to the first few tasks and random labels to remaining ones; and (c) systematically provide $+1$ labels to certain types of tasks and -1 to other types. Vuurens et al. (2011) recently provided real-world empirical evidence of similar worker strategies. In addition, workers may be malicious and may adopt sophisticated strategies for explicitly altering inferred labels of tasks. For instance, Wang et al. (2014) showed that malicious crowdsourcing campaigns, called "crowdturfing", are increasing in popularity in both dedicated and generic crowdsourcing websites (Motoyama et al., 2011; Wang et al., 2012). Further, evidence indicates the presence of malicious users in online content rating systems (such as Digg, Amazon, and Yelp) in which users can choose items to rate and the ratings are public. Specifically, users have been observed to explicitly alter the popularity of advertisements and phishing articles (Tran et al., 2009) and collaboratively target products on Amazon (Mukherjee et al., 2012).

This paper focuses on the problem of explicitly identifying *unreliable* and *adversarial* workers in crowdsourced labeling tasks by using only the labels provided by workers as an input. We consider a general crowdsourcing setting in which users/workers provide labels to items/tasks. The setting may be a crowdsourced classification application (such as Mechanical Turk) in which labels are collected for tasks by assigning [2] them to workers; or a public crowdsourcing system (such as Digg, Amazon, or Yelp) in which users provide labels/ratings to a collection of items they choose. For brevity, we use the generic terms "worker" and "task" for both types of applications. We make the following assumptions. The tasks have binary true labels in the set $\{-1, +1\}$; for cases in which the notion of true task labels is subjective, we consider it to be the population's majority opinion. We distinguish between two types of workers: *honest* and *adversarial*. The population of workers is mostly honest, with adversaries comprising a "small" fraction. Honest workers provide responses according

1. Note that the one-coin model cannot capture this strategy, but the more general two-coin model (Raykar and Yu, 2012) can
2. Workers can still choose from among assigned tasks; however, the assignment can be done to ensure that the graph representing workers' assignment to tasks has particular structures—see Section 4.1.

to a well-defined probabilistic model, say, M (e.g. the one-coin model introduced above), and therefore, they *can make mistakes*. However, the model M is not known. Motivated by the presence of non-random worker strategies, we go beyond standard probabilistic models and consider a much broader class of *adversarial* worker strategies. Specifically, adversaries adopt strategies different from those of honest workers, whether probabilistic or not, that is, their responses are not generated according to model M. Further, different adversaries may adopt distinct strategies.

Our work differs from most prior literature (Dawid and Skene, 1979; Smyth et al., 1995; Whitehill et al., 2009; Raykar et al., 2010; Welinder et al., 2010; Ghosh et al., 2011; Karger et al., 2011; Liu et al., 2012; Zhou et al., 2012; Dalvi et al., 2013; Zhang et al., 2014) that primarily focused on designing label aggregation algorithms to maximize the accuracy of recovered task labels. Only a few studies have explicitly focused on identifying unreliable workers based on their responses; these either relied on access to true task labels (Snow et al., 2008; Downs et al., 2010; Le et al., 2010) or did not assume any form of adversarial behavior (Vuurens et al., 2011; Raykar and Yu, 2012; Hovy et al., 2013). We aim to identify unreliable and adversarial workers using *only* their responses and do not assume access to any true task labels. Furthermore, to the best of our knowledge, we are unaware of previous studies that have addressed the problem of identifying adversarial workers who can adopt *arbitrary* strategies in crowdsourcing systems. Refer to Section 1.2 for further discussion on this aspect.

For the above setting, we design a scoring algorithm that computes "reputation scores" for workers to indicate the degree to which their labeling patterns are adversarial. We base our algorithms on the intuition that as the population is mostly honest and adversaries' labeling patterns differ from those of honest workers, adversary labeling patterns should be statistical outliers. The reputation score then indicates the degree to which a worker's labeling pattern is a statistical outlier. The adversaries identified by our algorithms may be discarded or processed separately depending on the application. Section 5 shows that discarding adversary labels can enable standard label aggregation algorithms to infer true task labels more accurately.

We note that the problem of identifying unreliable honest workers and adversaries from only their responses is nontrivial, especially because we cannot access true labels for any of the tasks. In particular, even in the simple and commonly observed case where adversaries always provide the label $+1$, the natural approach of classifying workers who only give $+1$ labels as adversaries can have arbitrarily bad performance (see Lemma 2). Furthermore, because we do not restrict the adversary strategy in any way, differentiating them from honest workers can be difficult. In fact, we show (see Theorem 14) that by carefully choosing their responses, *sophisticated* adversarial workers can render themselves *indistinguishable* from honest ones and ensure that the true labels of some fraction of tasks cannot be inferred better than a random guess.

1.1 Main contributions

Our work makes algorithmic, theoretical, and empirical contributions: [3]

3. This work expands on the results described in a previous version (Jagabathula et al., 2014)

Algorithmic contributions. Our main algorithmic contribution is a reputation algorithm that is designed to identify outlier labeling patterns. This algorithm takes as inputs the set of workers, set of tasks, and binary labels provided by each worker. Each worker may label only a subset of the tasks. Because the algorithm makes no specific assumptions on the worker labeling strategies, it identifies outliers by *penalizing* workers for the number of "conflicts" they are involved in. Specifically, suppose each task receives both $+1$ and -1 labels. For each task t_j, the algorithm maintains the number d_j^+ of $+1$ labels, number d_j^- of -1 labels, and a penalty budget of 2. Intuitively, if $d_j^+ > d_j^-$, then the worker assigning -1 to t_j is "more" of an outlier than a worker assigning label $+1$. Accordingly, the algorithm makes the following decisions: (a) how much of the penalty budget to allocate to each worker for each task and (b) how to aggregate the penalties allocated to each worker to arrive at the final reputation score.

We propose two algorithms that differ in how they make these two decisions. The first algorithm (Algorithm 1) performs a "soft" assignment: it allocates a penalty of $1/d_j^+$ (resp. $1/d_j^-$) to every worker who has provided the label $+1$ (resp. -1) to task t_j and computes the net penalty as the average of the penalties allocated across all tasks assigned to a worker. [4] The second algorithm (Algorithm 2) performs a "hard" assignment: instead of spreading the penalty score of 1 over all workers who provide the label $+1$ (resp. -1), it identifies one "representative" worker among the workers who provide the label $+1$ (resp. -1) to task t_j and allocates the entire penalty of 1 to the representative worker. The net penalty for the worker is computed by summing the accrued penalties across all assigned tasks. If the representative worker is chosen uniformly at random, then a worker who agrees with the majority is less likely to be chosen and therefore less likely to receive the penalty. However, we show that it is more appropriate to choose the representative worker in a "load-balanced" fashion by using the concept of *optimal semi-matching* (Harvey et al., 2003), where the semi-matching is defined on the bipartite graph between the workers and tasks, in which every worker is connected to the tasks that it has labeled. Both penalty assignments are based on the intuition that when there are more honest workers than adversaries, the former are more likely to agree with the majority and therefore receive lower penalties.

Theoretical contributions. We analyze our algorithms under three settings: (a) there are no adversaries and honest workers adopt the one-coin model; (b) honest workers adopt the one-coin model and adversaries, the Uniform strategy in which they label all assigned tasks $+1$; and (c) honest workers adopt the spammer-hammer model (Karger et al., 2011) and the adversaries are *sophisticated*, having infinite computational capacity and knowledge of honest workers' labels. For the first two settings, we derive guarantees for the soft-penalty algorithm and show that these guarantees extend to a random, normalized approximation of the hard-penalty algorithm; we analyze this approximation because the hard-penalty algorithm is not analytically tractable in these settings. For the last setting, we show that the hard-penalty algorithm is robust to sophisticated adversary strategies but the soft-penalty algorithm is vulnerable. Specifically, the example at the end of Section 3.1 shows an instance of an adversary strategy for which the soft-penalty algorithm misclassifies

4. We do not explicitly define the reputation score. It can be interpreted as the inverse of the net penalty—higher the penalty, the lower is the reputation, and vice-versa.

all adversaries as honest. We also show (see Lemma 13) that existing label aggregation algorithms can perform very poorly in the presence of sophisticated adversaries.

For the first two settings, we analyze our algorithms under a standard probabilistic model for crowdsourcing in which there are n workers; the worker-task assignment graph is (l, r)-regular, that is, each worker labels l tasks and each task is labeled by r workers; and the population is described by three parameters: fraction q of honest workers, fraction γ of tasks with $+1$ true labels, and average reliability μ of honest workers under the one-coin model. Recall that the reliability of an honest worker is defined as the probability of providing the correct response for any given task. This setup best represents the setting of crowdsourced classification in which tasks may be assigned according to an (l, r)-regular graph. An (l, r)-regular graph is analytically tractable, and Karger et al. (2014) showed that it achieves order-optimal performance (with respect to the best combination of task assignment and inference algorithm) when given a task assignment budget. We derive the expected penalties received by honest and adversarial workers and bound the asymptotic *error rate* or *misclassification rate* of a threshold classifier: given a penalty threshold θ, the threshold classifier classifies a worker as honest if its penalty is less than or equal to θ and as adversarial otherwise. Then, the misclassification rate is defined as the expected fraction of errors made by this classifier. This is non-trivial, especially because the natural approach that classifies all workers that provide only $+1$ labels as adversaries can have a misclassification rate arbitrarily close to 1 (see Lemma 2).

For the last setting, we make no assumptions on the structure of the worker-task assignment graph and derive performance guarantees as a function of the graph structure. This setup is reflective of public crowdsourced settings in which workers choose which tasks to label and the assumption of sophisticated adversaries is most relevant.

Formally, we establish the following results:

1. *No adversaries.* We show that the penalties assigned by the soft-penalty algorithm are consistent with worker reliabilities in the one-coin model—the higher the reliability, the lower is the penalty (see Theorem 3). When $l = \log n$ and r is constant, we show that the misclassification rate scales as $O(1/n^{2\varepsilon^2})$ for some small enough $\varepsilon \in (0, \frac{1}{\sqrt{2}})$ as $n \to \infty$, where a worker is said to be misclassified if it has high reliability (exceeding a threshold) but is classified as adversarial (see Theorem 6). In other words, our algorithm classifies all highly reliable workers as honest as $n \to \infty$. For this, the number of labels collected from each worker must tend to infinity, but only logarithmically in n.

2. Uniform *adversaries.* We derive necessary and sufficient conditions under which the soft-penalty algorithm assigns lower expected penalties to honest workers than to adversaries (see Theorem 4). Our conditions essentially require a sufficient majority of honest workers; specifically, for fixed μ and γ, we need the fraction of honest workers $q > \max\left\{1/2, h_\mu^{-1}\left(\gamma/(1-\gamma)\right)\right\}$, where $h_\mu^{-1}(\cdot)$ is an increasing function (see Theorem 4). This result shows that as γ increases, it becomes harder to separate honest workers from adversaries. This is because as the proportion γ of tasks with true labels $+1$ increases, the adversaries (who always provide the label $+1$) become more accurate, making it harder to distinguish them from honest workers. Further, when $l = \log n$ and r is constant, we show that the misclassification rate scales as

$O(1/n^{2\varepsilon^2})$ for adversaries and as $F(\tilde{\mu}) + O(1/n^{2\varepsilon^2})$ for honest workers, for some $\tilde{\mu} < \mu$ and small enough $\varepsilon > 0$ as $n \to \infty$ (see Theorem 10). Here, $F(\cdot)$ is the cumulative distribution function (CDF) of honest workers' reliabilities. Our result essentially shows that asymptotically, we correctly classify all adversaries and honest workers with above average reliabilities. We argue that the presence of the term $F(\tilde{\mu})$ is necessary because it is difficult to distinguish low-reliability honest workers from adversaries.

3. *Sophisticated adversaries.* We suppose that the goal of sophisticated adversaries is to maximize the number of tasks they *affect*, that is, cause to receive labels different from those they would have received otherwise. When honest workers are perfectly reliable, we show that k sophisticated adversaries can render themselves indistinguishable from k honest workers, so that a random guess misclassifies workers with probability (1/2). We use this to provide a lower bound on the minimum number of tasks that k adversaries can affect (the true label cannot be inferred better than a random guess) *irrespective* of the label aggregation algorithm used to aggregate the worker labels (as long as it is agnostic to worker/task identities). The bound depends on the graph structure between honest workers and tasks (see Theorem 14 for details). Our result is valid across different labeling patterns and a large class of label aggregation algorithms, and therefore, it provides *fundamental* limits on the damage that k adversaries can cause. Furthermore, we propose a label aggregation algorithm (Algorithm 3) utilizing worker reputations computed by the hard-penalty algorithm and prove the existence of an upper bound on the worst-case number of affected tasks (see Theorem 17), under the assumption that honest workers adopt the popular spammer-hammer model. This combined with the result of Theorem 14 shows that our proposed label aggregation algorithm is optimal (up to a constant factor) in recovering the true labels of tasks.

Empirical contributions. We conducted two numerical studies to demonstrate the practical value of our methods (see Section 5). The first study illustrates a concrete application of our methods. With five real-world crowdsourcing datasets, it shows that *discarding* the labels of adversaries identified by our methods allows standard label aggregation algorithms to infer true task labels more accurately. These improvements are demonstrated for the following six label aggregation algorithms: (a) simple majority, (b) expectation-maximization (EM) for the two-coin model (Raykar and Yu, 2012), (c) KOS (Karger et al., 2011), (d) a normalized variant of KOS [5], (e) spectral EM (Zhang et al., 2014), and (f) regularized minimax conditional entropy (Zhou et al., 2015). Our results show that by removing up to 10 workers, our methods can improve predictive accuracy by 9.8% on average. These improvements suggest that the label patterns of discarded workers do not conform to standard probabilistic models.

The second study is designed to complement our theoretical analysis. By using synthetic data, we show that both soft- and hard-penalty algorithms successfully identify Uniform adversaries who label all assigned tasks +1 and low-reliability honest workers when the worker-task assignment graph has a power-law degree distribution. These degree distributions commonly arise in crowdsourcing systems when workers organically choose the

5. This variant was designed to capture non-uniform worker and task degrees because the standard KOS algorithm is designed to operate on (l, r)-regular worker-task assignment graphs.

tasks to label, with some workers labeling many tasks and some tasks receiving many labels (Franklin et al., 2011). This study also offers insights into the settings under which the soft- or hard-penalty algorithm is appropriate; specifically, these algorithms are more appropriate when the adversaries have lower and higher degrees, respectively.

1.2 Related Work

Our work is part of the literature on crowdsourcing that proposes statistical techniques to exploit the redundancy in collected labels to simultaneously infer the latent reliabilities of workers and true task labels. In particular, our work is related to three broad streams. The first stream focuses on "crowdsourced classification", namely, inferring underlying true labels of tasks when workers adopt specific probabilistic labeling strategies. Our reputation algorithm can work in conjunction with any of these methods, possibly by filtering out low reputation workers. The second stream proposes methods to explicitly filter out low-reliability workers; it is similar in spirit to our approach. The third stream focuses on methods to address sophisticated attacks in online settings; it is related to our treatment of sophisticated adversaries.

Crowdsourced classification. The literature on crowdsourced classification is vast. Most studies are based on the worker model proposed by Dawid and Skene (1979), which is a generalization of the one-coin model to tasks with more than two categories. The standard solution is to use the expectation-maximization (EM) algorithm (or its variants) to estimate worker reliability parameters and true task labels (Smyth et al., 1995; Raykar et al., 2010). The methods proposed in Liu et al. (2012) and Chen et al. (2013) take a Bayesian approach by assuming different priors over the worker reliability parameters. Whitehill et al. (2009) included task difficulty as an additional parameter in the model, and Welinder et al. (2010) studied a model with multi-dimensional latent variables for each worker, such as competence, expertise, and bias. Zhou et al. (2012, 2015) introduced a natural generalization of the Dawid-Skene model that captures tasks with differing difficulties and proposed a minimax entropy based approach which works well in real datasets. Although most of these approaches show improved performance on real-world datasets, they offer no theoretical guarantees on the resulting estimates of true task labels and model parameters.

From a theoretical perspective, the crowdsourced classification problem has been studied in two distinct regimes: *dense* and *sparse*. In the *dense* regime, it is assumed that each worker has a certain probability of labeling each task. As the problem size (or number of workers) increases, each task receives an increasing number of responses; therefore, the true labels of all tasks are eventually identified correctly (with high probability). The theoretical analysis therefore focuses on identifying the rate at which the algorithm estimates converge to the true task labels under different settings. The dense regime was first studied by Ghosh et al. (2011), who proposed a spectral method to infer task labels. More recently, Gao and Zhou (2016) studied the minimax optimal error rate of a projected EM algorithm under the one-coin model, and Li and Yu (2014) provided upper bounds on the error rate of weighted majority voting algorithms for the Dawid-Skene model. Zhang et al. (2014) showed that the EM algorithm for the Dawid-Skene model achieves the optimal convergence rate when initialized using a spectral method.

In the *sparse* regime, each task is assigned a "small" number of workers, that is, of size $O(1)$, so that the accuracy does not increase with problem size. Karger et al. (2014) were the first to analyze this scenario; they proposed an iterative message-passing algorithm for estimating true task labels as well as a task assignment scheme that minimizes the total price that must be paid to achieve an overall target accuracy. They showed that their algorithm is optimal by comparing it against an oracle estimator that knows the reliability of every worker. Dalvi et al. (2013) proposed methods based on singular-value decomposition (SVD) and analyzed the consistency of their estimators for the one-coin model. Recently, Khetan and Oh (2016) analyzed the "Generalized Dawid-Skene Model" introduced by Zhou et al. and showed that spectral approaches achieve near-optimal performance, whereas Ok et al. (2016) proved that belief propagation (BP) is optimal, that is, it matches the performance of the MAP estimator, under the one-coin model when each worker is assigned at most two tasks. In our theoretical analysis, we focus on this regime. The key distinction of our work from previous studies is that we focus on characterizing the misclassification rate of our algorithm in classifying workers, as opposed to the accuracy of recovering true task labels.

Detecting Unreliable/Adversarial workers. Some studies have aimed to explicitly detect and/or remove unreliable workers based on observed labels. One approach is to use "gold standard" tasks, that is, tasks whose true labels are already known, to identify low-reliability workers (Snow et al., 2008; Downs et al., 2010; Le et al., 2010). However, accessing true task labels can be difficult and might involve additional payment. In this work, we do not assume access to any gold standard tasks and identify adversarial workers based only on the provided labels. Vuurens et al. (2011) defined scores customized to specific adversary strategies to identify and remove them. Similarly, Hovy et al. (2013) modeled the worker population as consisting of two types—those who always provide the correct label and spammers who provide uniformly random labels—and estimated each worker's trustworthiness by using the observed labels. In contrast to these studies, we allow adversaries to adopt arbitrary strategies. Ipeirotis et al. (2010) proposed a method for quantifying worker quality by transforming the observed labels into soft posterior labels based on the estimated *confusion matrix* (Dawid and Skene, 1979). Similar to our work, their approach computes an expected cost for each worker, where the higher the cost, the lower is the worker's quality. Raykar and Yu (2012) proposed an empirical Bayesian algorithm to eliminate workers whose labels are not correlated with the true label (called *spammers*), and estimated consensus labels from the remaining workers. Both of these works rely on the Dawid-Skene model, whereas our algorithms do not assume knowledge of the probabilistic model used by workers. Some studies have tried to quantify the price of having adversarial workers under some restricted settings. Ghosh et al. (2011) (in the dense regime) and Karger et al. (2013) (in the sparse regime) considered malicious workers who can collude and provide arbitrary responses to degrade the performance of the aggregation algorithms and showed that their approaches are robust to manipulation by a small constant fraction of such adversaries. However, both these works assume specific structures on the worker-task assignment graph and do not consider adversaries who can adapt their responses based on the labels submitted by honest workers. Our analysis considers *arbitrary* worker-task assignment graphs, and we allow adversaries to choose their labels based on observed honest workers' responses.

Sybil attacks. Finally, our work is also broadly related to the rich literature on identifying *Sybil* identities in online social networks. Most such schemes (Yu et al., 2006, 2008; Danezis and Mittal, 2009; Tran et al., 2011; Viswanath et al., 2012) use the graph (or trust) structure between users to limit the corruptive influences of *Sybil attacks* (see Viswanath et al., 2010 for a nice overview). In our context, there is no information about the network structure or trust relationships between workers, and because most crowdsourcing tasks involve some form of payment, it is harder to launch Sybil attacks by forging financial credentials like credit cards or bank accounts.

2. Setup

We consider the following broad setting. There is a set $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ of m tasks such that each task t_j is associated with a latent ground-truth binary label $y_j \in \{-1, +1\}$. We elicit binary labels for these tasks from a set $W = \{w_1, w_2, \ldots, w_n\}$ of n workers. Each worker typically labels only a subset of the tasks, and we generically say that the subset of tasks is *assigned* to the worker. We represent this assignment by using a bipartite graph $\mathcal{B} = (W \cup \mathcal{T}, E)$ with workers on one side, tasks on the other side, and an edge $(w_i, t_j) \in E$ indicating that task t_j is assigned to worker w_i. We call $\mathcal{B}$ the *worker-task assignment graph* and suppose that the assignment is pre-specified.

Each worker w_i provides a binary response [6] $w_i(t_j) \in \{-1, +1\}$ for each task t_j assigned to it. We encode the responses as the *response matrix* $\mathcal{L} \in \{-1, 0, +1\}^{|W| \times |\mathcal{T}|}$ such that $\mathcal{L}_{ij} = w_i(t_j)$, for all $1 \leq i \leq n$ and $1 \leq j \leq m$, where we set $w_i(t_j) = 0$ for any task t_j not assigned to worker w_i. $W_j \subseteq W$ denotes the set of workers who labeled task t_j and $\mathcal{T}_i \subseteq \mathcal{T}$, the set of tasks assigned to worker w_i. Let d_j^+ (resp. d_j^-) denote the number of workers labeling task t_j as $+1$ (resp. -1).

Worker model. We assume that the population of workers comprises two disjoint classes: *honest* and *adversarial*. That is, $W = H \cup A$ with $H \cap A = \emptyset$, where H is the set of honest workers and A, the set of adversarial workers. The class memberships of the workers are latent, so we do not know whether a worker is honest. Honest workers provide (noisy) labels according to an unknown but explicit probabilistic model (such as the one-coin model). Adversarial workers are those whose labeling strategy does not conform to this probabilistic model; they can adopt arbitrary (deterministic or probabilistic) strategies. If honest workers adopt the one-coin model, example adversary strategies include (a) the *uniform strategy*, in which the worker arbitrarily provides uniform responses $+1$ or -1 to all assigned tasks irrespective of the true label; (b) *smart strategy*, in which the adversary is smart and chooses the uniform label in accordance with the population prevalence of the true labels so that if more than 50% of tasks are a priori known to have label -1, then the worker chooses -1 as the uniform label and vice-versa; or (c) *sophisticated strategy*, in which the worker adopts strategies specifically designed to cause the maximum "damage" (see Section 4.2 for details).

Most existing works only focus on honest workers, whereas our approach also considers adversarial workers. Furthermore, our definition of "adversary" is intentionally broader than common definitions to accommodate a wide range of labeling strategies. In fact, some of the abovementioned example adversary strategies may be accommodated by exten-

6. We use the terms "label" and "response" interchangeably.

sions of the one-coin model; for example, the uniform strategy is captured by the two-coin model (Raykar and Yu, 2012), and the smart strategy can be accommodated by allowing worker reliability parameters to depend on the population prevalence of the task labels. While such case-by-case extensions are feasible in theory, they do not extend to general adversary strategies, including sophisticated strategies specifically designed to inflict the maximum "damage".

Given the broad definition of adversaries, our approach is to design a general algorithm to *identify* adversarial workers. The adversaries identified using our algorithm may be filtered out or investigated further, depending on the application. Specifically, our objective is to solve the following problem:

Problem 1 *Given a set of workers $W = H \cup A$, tasks $\mathcal{T}$, and response matrix $\mathcal{L}$, identify the subset of adversarial workers A.*

We describe a reputation-based algorithm that only relies on the response matrix $\mathcal{L}$ to detect adversaries. The algorithm relies on detecting workers whose labeling patterns are statistical *outliers* among the population of workers.

3. Reputation Algorithms

We now describe the proposed algorithm for identifying adversarial workers given the response matrix $\mathcal{L}$. We suppose that no side information is available on the workers' identities (e.g., a social network or worker-level demographic information), and therefore, the algorithm must solely rely on the response patterns given by workers. Our approach is to compute a "reputation" or "trust" score for each worker as a measure of the degree to which their response pattern is a statistical outlier or an anomaly. Workers with low reputation scores are significant outliers and are identified as adversaries.

To compute a worker's reputation, the algorithm relies on the number of *conflicts* the worker is involved in. A worker is involved in a conflict if its response to an assigned task is in disagreement with those of other workers. Note that tasks with a consensus opinion, that is, those having all $+1$ or -1 labels, do not provide any *discriminative* information about the workers who labeled the task. In other words, we cannot distinguish between honest and adversarial workers from just this specific task. Therefore, we focus on tasks that lack consensus, that is, those having a mix of $+1$ and -1 labels. We call this subset of tasks the *conflict set* $\mathcal{T}_{cs}$, and workers who respond to tasks in this set are all involved in conflicts. A conflict typically indicates the presence of low-reliability honest workers (who tend to make mistakes) or adversaries. In the ideal case, when all honest workers are perfectly reliable, a conflict necessarily means the presence of an adversarial worker. In this case, the number of conflicts a worker is involved in can serve as a rough indicator of the possibility of this worker being an adversary. However, when honest workers are not perfect and make mistakes, a conflict indicates only a *chance* of the presence of an adversary. Then, simply counting the number of conflicts may over-penalize honest workers who label a large number of tasks.

To overcome this issue, we propose two penalty allocation techniques, resulting in two variants of our algorithm: (a) *soft-penalty* and (b) *hard-penalty*.

3.1 Soft Penalty

In the *soft-penalty* algorithm (see Algorithm 1), for any task t_j in the conflict set, we allocate a penalty of $1/d_j^+$ and $1/d_j^-$ to all workers who provide the label $+1$ and -1 for t_j, respectively. Then, for each worker, we compute the *net penalty* by averaging the penalties across all assigned (conflict) tasks.

The above allocation of penalties implicitly rewards agreements among worker responses by making the penalty inversely proportional to the number of other workers that agree with a worker. In particular, if a worker agrees with the majority opinion on some task, then it is allocated a lower penalty than a worker who disagrees with the majority. Further, averaging normalizes for the number of tasks labeled by any worker. The algorithm relies on the following intuition for allocating penalties: assuming the average reliability of honest workers to be $> \frac{1}{2}$, we expect that honest workers provide the correct response to the assigned tasks on average. Furthermore, because there are more honest workers than adversaries, we expect the majority response to be the same as the true label of the task for most tasks. Therefore, we expect that the above allocation of penalties assigns lower penalties to high-reliability honest workers and higher penalties to low-reliability honest *and* adversarial workers. We formalize this intuition in Section 4, where we prove theoretical guarantees for the soft-penalty algorithm. We show that the soft-penalty algorithm performs well in identifying low-reliability honest workers as well as adversarial workers employing deterministic strategies (see Theorems 3, 4, 6, and 10). Our results demonstrate the asymptotic consistency of the soft-penalty algorithm in identifying adversaries under standard assumptions on the structure of the worker-task assignment graph.

Although the soft-penalty algorithm can successfully identify adversarial workers adopting certain types of strategies, its performance depends on the complexity of these strategies. If adversarial workers are non-colluding and adopt non-deliberate strategies, then the soft-penalty algorithm can identify them from the observed responses. However, this algorithm could be manipulated by more sophisticated adversaries who can collude together and adapt their labeling strategy to target certain tasks to lower their penalty scores. In particular, the soft-penalty algorithm treats each task in isolation when assigning penalties; therefore, it is susceptible to attack by determined adversaries who can cleverly decide their responses based on honest workers' labels and the structure of the worker-task assignment graph to cause maximum "damage". For example, suppose that the subgraph of $\mathcal{B}$ between honest workers and tasks is r-right regular, that is, each task receives labels from exactly r honest workers (such graphs are commonly used in practice, see Karger et al., 2014 and Ok et al., 2016), and all honest workers are perfectly reliable. Now, suppose that there are $k > r$ adversaries and that each adversary provides the incorrect response to *all* tasks. Then, every task has r correct responses, all provided by honest workers, and k incorrect responses, all provided by adversaries, resulting in net penalties of $1/r$ for each honest worker and $1/k$ for each adversary (note that the degree of the workers does not affect the net penalty because the penalty received from each task is the same). Because $k > r$, adversaries receive lower penalties than do honest workers. Therefore, filtering out k workers with the highest penalties will always filter out honest workers. Furthermore, natural aggregation algorithms (such as simple majority or weighted majority with penalties as weights) result

in incorrect labels for all tasks (see Lemma 13). In fact, for such worst-case adversaries, we can establish the following result:

Theoretical Result (Refer to Theorem 14). *Given any collection of honest worker responses, there exists an adversary strategy that can achieve a lower bound on the fraction of tasks whose true labels cannot be inferred correctly (better than a random guess) by* ***any*** *label aggregation algorithm that is agnostic to the worker and task identities.*

Section 4.2.2 provides a formal description of the result. The proof relies on the fact that sophisticated adversaries can render themselves *indistinguishable* from honest workers by carefully choosing their responses. This shows that identifying such worst-case adversarial workers can be difficult. To deal with such adversarial behavior, we use the *hard-penalty* algorithm.

Algorithm 1 SOFT PENALTY

1: **Input:** W, $\mathcal{T}$, and $\mathcal{L}$

2: For every task $t_j \in \mathcal{T}_{cs}$, allocate penalty s_{ij} to each worker $w_i \in W_j$ as follows:

$$s_{ij} = \begin{cases} \frac{1}{d_j^+}, & \text{if } \mathcal{L}_{ij} = +1 \\ \frac{1}{d_j^-}, & \text{if } \mathcal{L}_{ij} = -1 \end{cases}$$

3: **Output:** Net penalty of worker w_i:

$$pen(w_i) = \frac{\sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij}}{|\mathcal{T}_i \cap \mathcal{T}_{cs}|}$$

Algorithm 2 HARD PENALTY

1: **Input:** W, $\mathcal{T}$, and $\mathcal{L}$

2: Create a bipartite graph $\mathcal{B}^{cs}$ as follows: (i) Each worker $w_i \in W$ is represented by a node on the left (ii) Each task $t_j \in \mathcal{T}_{cs}$ is represented by two nodes on the right, t_j^+ and t_j^- (iii) Add the edge (w_i, t_j^+) if $\mathcal{L}_{ij} = +1$ or edge (w_i, t_j^-) if $\mathcal{L}_{ij} = -1$

3: Compute an optimal semi-matching $\mathcal{M}$ on $\mathcal{B}^{cs}$

4: **Output:** Net penalty of worker w_i: $pen(w_i) = deg_{\mathcal{M}}(w_i)$

3.2 Hard Penalty

To deal with sophisticated adversaries, we propose a *hard* penalty allocation scheme (Algorithm 2) in which the penalty allocation for a particular task takes into account the structure of the worker-task assignment graph and the responses of other workers on *all* other tasks. In particular, instead of distributing the penalty evenly across all workers that respond to a given task, this algorithm chooses two "representative" workers to penalize for each conflict task: one each among those who provide the label $+1$ and -1. The representative workers are chosen in a load-balanced manner to "spread" the penalty across all workers and thereby avoid over-penalizing workers who provide labels for a large number of tasks. The net penalty of each worker is the sum of penalties accrued across all (conflict) tasks assigned to this worker. Intuitively, such a hard allocation of penalties will penalize workers with higher degrees (i.e. large number of assigned tasks) and many conflicts (who are potential worst-case adversaries), thereby leading to a low reputation.

To choose representative workers in a load-balanced fashion, we use the concept of *optimal semi-matchings* (Harvey et al., 2003) in bipartite graphs. For a bipartite graph $G = (V_1 \cup V_2, E)$, a *semi-matching* in G is a set of edges $M \subseteq E$ such that each vertex in V_2 is incident to exactly one edge in M (note that vertices in V_1 could be incident to

multiple edges in M). A semi-matching generalizes the notion of matchings on bipartite graphs. The optimal semi-matching is the semi-matching with the minimum *cost*. We use the common degree-based cost function defined as follows: for each $u \in V_1$, let $deg_M(u)$ denote the *degree* of u, that is, the number of edges in M that are incident to u, and let $cost_M(u)$ be defined as

$$cost_M(u) := \sum_{i=1}^{deg_M(u)} i = \frac{deg_M(u) \cdot (deg_M(u) + 1)}{2}$$

Then, an *optimal semi-matching* is one that minimizes $\sum_{u \in V_1} cost_M(u)$. Intuitively, an optimal semi-matching *fairly* matches V_2 vertices across V_1 vertices such that the "load" on any V_1 vertex is minimized. The above notion of cost is motivated by the load balancing problem for scheduling tasks on machines. Specifically, consider a set of unit-time tasks T and a set of machines P. Suppose that each task t can be processed on a subset of machines; this can be specified as a bipartite graph between T and P. On any given machine, the tasks are executed one after the other in series. An optimal semi-matching can be thought of as an assignment of tasks to the machines such that the *flow-time*, that is, average completion time of a task, is minimized. See (Harvey et al., 2003) for more details.

To determine the representative workers for each task, we compute the optimal semi-matching in the following augmented worker-task assignment graph: we split each task t_j into two copies, t_j^+ and t_j^-, and connect worker w_i to t_j^+ or t_j^- depending on whether this worker labeled the task $+1$ or -1, respectively. By definition, the optimal semi-matching yields two representative workers for each task t_j—one connected to t_j^+ and the other connected to t_j^-. As in the soft-penalty algorithm, we only consider conflict tasks when creating this augmented bipartite graph. The worker degrees in the optimal semi-matching then constitute their net penalties. Algorithm 2 describes the hard-penalty algorithm.

3.3 Connection between soft-penalty and hard-penalty algorithms

Although the hard- and soft-penalty algorithms appear different, the latter can be interpreted as a random, normalized variant of the former. Specifically, suppose we choose a random semi-matching M in the augmented worker-task assignment graph $\mathcal{B}^{cs}$, defined in Algorithm 2, and assign the penalty $deg_M(w_i)/deg_{\mathcal{B}^{cs}}(w_i)$ to worker w_i, where $deg_{\mathcal{B}^{cs}}(w_i)$ is the degree of worker w_i in $\mathcal{B}^{cs}$. When the random semi-matching is constructed by mapping each copy t_j^+ (or t_j^-) of task t_j uniformly at random to a worker connected to it, the probability that it will be mapped to worker $w_i \in W_j$ is equal to $1/deg_{\mathcal{B}^{cs}}(t_j^+)$ (or $1/deg_{\mathcal{B}^{cs}}(t_j^-)$), or equivalently, $1/d_j^+$ (or $1/d_j^-$). Therefore, the expected degree $\mathbb{E}[deg_M(w_i)]$ of worker w_i is equal to $\sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij}$, where $s_{ij} = 1/d_j^+$ if $\mathcal{L}_{ij} = +1$ and $1/d_j^-$ if $\mathcal{L}_{ij} = -1$. Because the degree $deg_{\mathcal{B}^{cs}}(w_i)$ of worker w_i is equal to $|\mathcal{T}_i \cap \mathcal{T}_{cs}|$, it follows that the expected penalty of worker w_i is equal to $\mathbb{E}[deg_M(w_i)]/deg_{\mathcal{B}^{cs}}(w_i) = \sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij} / |\mathcal{T}_i \cap \mathcal{T}_{cs}|$, which is exactly the penalty allocated by the soft-penalty algorithm. It thus follows that the expected penalties under the above random, normalized variant of the hard-penalty algorithm are equal to the penalties allocated by the soft-penalty algorithm. When all workers are assigned the same number of tasks, the expected penalty assigned by the random hard-penalty al-

gorithm is equal to the penalty assigned by the soft-penalty algorithm, but scaled by a constant factor.

With the above interpretation of the soft-penalty algorithm, it follows that the hard-penalty algorithm differs from the soft-penalty algorithm in two key aspects: it (a) does not normalize the penalties by degrees and (b) uses optimal semi-matchings as opposed to random semi-matchings. The absence of degree-based normalization of the penalties results in significant penalization of high-degree workers. The use of the optimal semi-matching results in a more balanced allocation of penalties by optimizing a global objective function. Both of these effects make the hard-penalty algorithm conservative and robust to sophisticated adversary strategies, as established theoretically in Section 4. The above connection also suggests that the random, normalized variant of the hard-penalty algorithm should have performance similar to that of the soft-penalty algorithm. We explore this aspect theoretically at the end of Section 4.1.

4. Theoretical Results

Our reputation algorithms are analytically tractable, and we establish their theoretical properties below. Our analysis is aimed at deriving the conditions under which our algorithms separate adversaries from honest workers. We use uppercase boldface letters (say, $\boldsymbol{X}$) to denote random variables, unless it is clear from the context. The proofs of all results are given in Appendix A.

4.1 Soft-penalty algorithm: common adversary strategies

First, we analyze the performance of the soft-penalty algorithm. We assume that honest workers adopt the one-coin model, where each worker w is characterized by parameter $\mu_w \in [0, 1]$ that specifies the probability that w provides the correct response for any task. We choose this model because it is the simplest non-trivial model to analyze, and it has been widely studied in existing literature (Ghosh et al., 2011; Dalvi et al., 2013; Karger et al., 2014; Zhang et al., 2014; Gao and Zhou, 2016). We focus on two settings: (a) the classical setting in which there are no adversaries ($A = \emptyset$) and (b) the setting in which adversaries adopt the Uniform strategy.

Definition 1 (Uniform strategy) *Every adversarial worker provides the same $+1$ response to all tasks assigned to it.*

We consider the Uniform strategy because of its ubiquity and simplicity. It is commonly observed in practice (Vuurens et al., 2011), and our analysis of real-world datasets (discussed in detail in Section 5) reveals that a significant fraction of workers ($\approx 30\%, 17\%$, and 9% of workers in the `stage2, task2`, and `temp` datasets, respectively) provide uniform labels for all assigned tasks. Note that it is not captured by the standard one-coin model, in which workers assign the label y_j or $-y_j$ to task t_j, where y_j is the true label. It can be adopted by both "lazy" and "smart" adversaries. Lazy workers adopt this strategy to maximize the number of tasks they label and the corresponding payment they obtain. Smart adversaries adopt this strategy if it is known a priori that the prevalence (or proportion) γ of tasks with true labels $+1$ is large, say, above 90%. For instance, medical images showing tumors contain a large proportion of benign ones and a correspondingly small proportion of malignant ones,

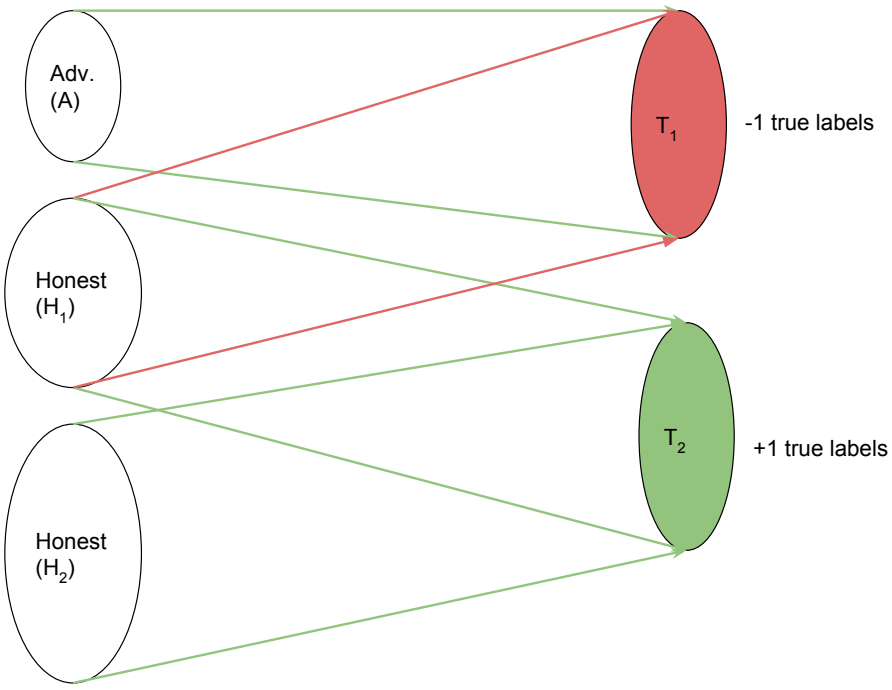

Figure 1: Example where filtering workers who give only +1 labels performs poorly

leading a "smart" worker to label all assigned images as benign without carefully considering each image. Therefore, the Uniform strategy actually comprises a spectrum of strategies of varying degrees of "smartness", with higher values of γ indicating smarter strategies.

A natural way to identify adversaries who adopt the Uniform strategy is to classify all workers who have labeled *all* assigned tasks +1 as adversaries. However, we show that this approach can have arbitrarily large *misclassification rate*, because it may misclassify almost all (asymptotic fraction approaching 1) perfectly reliable honest workers as adversarial.

Lemma 2 (Hardness of identifying Uniform adversarial workers) *Consider a simple binary classifier* $\hat{\boldsymbol{I}}_{\text{natural}}(\cdot)$ *as follows:*

$$\hat{\boldsymbol{I}}_{\text{natural}}(w_i) = \begin{cases} \textit{adversarial}, & \textit{if } w_i \textit{ gives all } +1 \textit{ labels} \\ \textit{honest}, & \textit{o.w.} \end{cases}$$

Then, the misclassification rate, that is, fraction of incorrectly classified workers, of the above classifier can be arbitrarily close to 1.

Proof Suppose the collection of tasks partitions into two sets, T_1 and T_2, consisting of tasks that have true labels -1 and $+1$, respectively (see Figure 1). The adversary A follows the Uniform strategy and labels all tasks in T_1 as $+1$. There are two groups of honest workers—H_1 and H_2—who are perfectly reliable, that is, they always provide the correct label. The workers in H_1 label all tasks in $T_1 \cup T_2$, and workers in H_2 label only the tasks in T_2. Now, because honest workers H_2 give only $+1$ labels, the classifier $\hat{\boldsymbol{I}}_{\text{natural}}$ misclassifies

all honest workers in H_2 as adversaries. In other words, we have

$$\frac{1}{|H_1 \cup H_2|} \sum_{h \in H_1 \cup H_2} \mathbf{1}\left[\hat{\boldsymbol{I}}_{\text{natural}}(h) \neq \textit{honest}\right] = \frac{|H_2|}{|H_1 \cup H_2|}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. Suppose H_2 comprises a large fraction of honest workers, that is, $|H_2| = (1-\rho)\,|H_1 \cup H_2|$ for some "small" $\rho > 0$. Then, it follows from the above equation that the misclassification rate of the natural classifier $\hat{\boldsymbol{I}}_{\text{natural}}(\cdot)$ is $1-\rho$, which can be arbitrarily close to 1. ■

The above lemma shows that the problem of identifying adversaries is nontrivial even for the Uniform strategy case, and our analysis below provides precise conditions under which we can identify such adversarial workers. In particular, for the the scenario outlined in the proof above, our soft-penalty algorithm assigns a penalty of $\frac{1}{|A|}$ to adversaries, $\frac{1}{|H_1|}$ to honest workers in H_1, and zero penalty to honest workers in H_2 (because they are not part of any conflicts). Consequently, as long as $|H_1| > |A|$, our algorithm correctly separates out honest workers from adversaries by choosing a threshold $\theta \in \left(\frac{1}{|H_1|}, \frac{1}{|A|}\right)$ and classifying all workers with penalty $> \theta$ as adversarial.

Because the performance of the algorithm depends on the specific crowdsourced classification instance (worker-task assignment graph, true task labels, reliabilities of honest workers), we conduct a probabilistic analysis under a natural generative model. For our analysis, we focus on worker-task assignment graphs $\mathcal{B}$ that are (l, r)-regular, in which each worker is assigned l tasks and each task is labeled by r workers. These assignment graphs are analytically tractable and have been shown by Karger et al. (2014) to achieve order-optimal performance (with respect to the best combination of task assignment and inference algorithm) when given a certain budget for task assignment. To generate the crowdsourcing instance, we use the probabilistic model of crowdsourced labeling proposed by Karger et al. (2014) but extended to incorporate adversarial workers:

Generative model. Suppose the fraction $q \in (0, 1]$ of honest workers, number of tasks m, number of workers n, worker degree $l > 1$, and task degree $r > 1$ are fixed. Let $\gamma \in [0, 1]$ denote the prevalence (or proportion) of tasks with true labels $+1$ and $F(\cdot)$, the cumulative distribution function (CDF) of the honest worker reliabilities under the one-coin model, with $\mu \in [0, 1]$ denoting the mean. Sample a crowdsourced classification instance as follows:

1. *Worker-task assignment graph:* Assign m tasks to n workers using the *configuration model*—take $n \cdot l$ half-edges for worker nodes and $m \cdot r$ half-edges for the task nodes, pick a random permutation of worker half-edges, and map them to task half-edges.

2. *True task labels:* For each task t_j, sample the true label $\boldsymbol{Y}_j \in \{-1, +1\}$ independently according to the Bernoulli distribution with $\Pr[\boldsymbol{Y}_j = +1] = \gamma$.

3. *Worker identities:* For each worker w_i, set its identity to *honest* with probability q and *adversarial* with probability $1 - q$.

4. *Honest worker reliabilities and responses:* If w_i is honest, sample its reliability $\boldsymbol{M}_i = \mu_i$ from the distribution $F(\cdot)$. For each task t_j assigned to w_i, set the response $w_i(t_j)$ to $\boldsymbol{Y}_j$ with probability μ_i and $-\boldsymbol{Y}_j$ with probability $1 - \mu_i$.

5. *Adversarial worker responses:* If w_i is adversarial, set the response $w_i(t_j) = +1$ for all tasks t_j assigned to w_i.

The above generative model may be justified as follows. First, the *configuration model* is a simple random construction to generate graphs that is popular in random graph literature (Bollobás, 2001). It may result in a graph with multi-edges (where two nodes are connected by more than one edge); however, the number of double-edges converges to a Poisson distribution with mean $(l-1)(r-1)/2$ (see Bollobás, 2001). Therefore, the proportion of nodes with multi-edges is $\approx lr/n$, which tends to zero as $n \to \infty$ as long as $l = o(n)$ and r is constant.

The model for true task labels, worker identities, and reliabilities may be justified by supposing that the m tasks $\mathcal{T} = \{t_1, \ldots, t_m\}$ are drawn from a "large" population of tasks with a prevalence γ of $+1$ tasks and workers are drawn from a "large" population with a proportion $1 - q$ of adversaries and a proportion q of honest workers, whose reliabilities have the distribution $F(\cdot)$. Then, the distributional assumptions for the true task labels, worker identities, and honest worker reliabilities will be met when the task assignment is randomized and there is no distinction between sampling with and without replacement because of the large population sizes. The model for generating honest worker responses is the standard one-coin model. See Karger et al. (2014) for a detailed discussion of the settings under which the above probabilistic model is reasonable.

For our theoretical analysis, we assume that non-conflict tasks (with all $+1$ or -1 labels) are *not* ignored/dropped for the purposes of penalty computation. This assumption makes the analysis less cumbersome and may be justified by noting that for a large enough r, the probability that a task will be non-conflict is low. Even if a task is non-conflict, we expect little impact from its inclusion because the penalty from this task will be $1/r$, which is negligible for large values of r. We also tested this assertion numerically and observed negligible differences in the performances of the two variants (with and without dropping high-degree non-conflict tasks) of the soft-penalty algorithm (see Section 5).

4.1.1 Analysis of expected penalties

We first analyze the expected penalties received by honest and adversarial workers under the abovementioned generative model and identify the conditions for population parameters q, μ, and γ under which honest workers receive lower expected penalties. Let $\mathbf{PEN}_i$ denote the penalty assigned by the soft-penalty algorithm to worker w_i; note that it is a random variable under the abovementioned generative model.

First, we focus on the classical setting in which there are no adversarial workers; therefore, $A = \emptyset$. We obtain the following result:

Theorem 3 (Reputations consistent with reliabilities) *When $q = 1$ (i.e., there are no adversarial workers) and $\mu > \frac{1}{2}$, we have*

$$\mathbb{E}\left[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_1\right] < \mathbb{E}\left[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_2\right] \iff \mu_1 > \mu_2$$

for any worker w_i.

Theorem 3 shows that the expected reputation scores are consistent with honest workers' reliabilities: as the honest worker's reliability decreases, the expected penalty increases,

that is, the expected reputation score decreases. As honest worker reliabilities capture their propensities to make mistakes, our algorithm flags workers who are prone to making mistakes, as desired. Consequently, filtering out low-reputation workers filters out workers with low reliabilities (we make this claim precise in Section 4.1.2 below).

Next, we consider the case in which $A \neq \emptyset$ and there is a fraction $1-q$ of workers who are adversarial and adopt the Uniform strategy. Let p_h and p_a denote the expected penalties that a worker receives conditioned on being honest and adversarial, respectively, that is,

$$p_h = \mathbb{E}\left[\mathbf{PEN}_i \mid w_i \text{ is honest }\right] \quad \text{and} \quad p_a = \mathbb{E}\left[\mathbf{PEN}_i \mid w_i \text{ is adversarial }\right]$$

Because of symmetry, these expectations do not depend on worker indices. Then, we obtain the following result:

Theorem 4 (Penalties under Uniform strategy) *When $q < 1$, $\mu > \frac{1}{2}$, and the adversaries adopt the* Uniform *strategy, we obtain*

$$p_h < p_a \iff \left(q\mu > \frac{1}{2} \quad \textit{and} \quad \frac{\mu}{1-\mu} \cdot h(\mu, q) > \frac{\gamma}{1-\gamma}\right)$$

where $h(\mu, q)$ is a strictly increasing function in q for a fixed μ, and it is defined as

$$h(\mu, q) = \frac{g(1-Q) - g(Q)}{g(P) - g(1-P)} \text{ where } P := q\mu + (1-q), Q := 1 - q\mu, \text{ and } g(x) := \frac{1-x^r}{r \cdot (1-x)}.$$

The above result reveals the conditions for the parameters q, μ, and γ under which the soft-penalty algorithm is successful in assigning lower penalties to honest workers than to adversaries.

To understand this, we first focus on the condition $q\mu > 1/2$. Note that because $\mu \leq 1$, this condition implies that the population must consist of more honest workers than adversaries ($q > 1/2$). This is because our algorithm is designed to identify "outlier" response patterns—those which deviate from the majority—and for adversaries to be declared outliers, they must necessarily be in the minority.

Furthermore, note that the necessary condition $\mu > 1/(2q)$ implies that for our algorithm to be successful, the average reliability μ of the honest workers must be "large enough"; specifically, it must exceed $\frac{1}{2q}$ (for a fixed q). To obtain an intuitive understanding of this condition (the proof is given in Appendix A.1.2), note the following. Consider a task with true label $+1$. Then, in expectation, there are $rq\mu$ honest workers and $(1-q)\cdot r$ adversaries who provide the response $+1$ and $rq \cdot (1-\mu)$ honest workers who provide the response -1. Now, the adversaries will agree with the majority if and only if $r \cdot (q\mu + (1-q)) \geq rq \cdot (1-\mu)$, that is, $\mu \geq 1 - 1/(2q)$. Similarly, when the true label of a task is -1, then in expectation, there are $r \cdot (q \cdot (1-\mu) + (1-q))$ workers providing $+1$ label and $rq\mu$ workers providing -1 label. Again, the adversaries will be in the majority in expectation if and only if $\mu \leq 1/(2q)$. It thus follows that if $\mu \in [1 - \frac{1}{2q}, \frac{1}{2q}]$, then the adversaries are always in majority in expectation, and therefore, they will receive a lower penalty. Because $\mu > 1/2$ and $1 - \frac{1}{2q} \leq 1/2$, a necessary condition for the worker to receive a lower penalty when being honest is therefore $\mu > \frac{1}{2q}$, that is, $q\mu > \frac{1}{2}$.

Now assuming that the first condition is met, that is, $q\mu > 1/2$, we focus on the second condition: $\frac{\mu}{1-\mu} \cdot h(\mu, q) > \frac{\gamma}{1-\gamma}$. When $\gamma = 1$, this condition is not met (unless $\mu = 1$), and therefore, honest workers receive higher expected penalties than adversaries. This is because if all tasks have a true label $+1$, then in expectation, there is a fraction $q\mu + 1 - q > 1/2$ (because $q\mu > 1/2$) of workers providing the label $+1$ to each task, implying that the adversaries always agree with the majority label for each task. As a result, our algorithm filters out honest workers; however, it must be noted that adversaries actually have higher (perfect) reliabilities in this special case. Similarly, when $\mu = 1$, that is, honest workers are perfectly reliable, the condition is always met because honest workers are in the majority at each task (in expectation); specifically, when the true label is $+1$, all responses are $+1$, and when the true label is -1, all honest workers (a fraction $q > 1/2$) provide the label -1.

Next, we investigate the performance of our algorithm as the adversary strategies become "smarter". As noted above, the Uniform strategy comprises a spectrum of strategies of varying degrees of "smartness", with higher values of γ indicating smarter strategies.

Corollary 5 (Penalties under smarter adversary strategies) *Suppose $\mu > \frac{1}{2}$ is fixed and $q\mu > \frac{1}{2}$. Then, it follows that*

$$p_h < p_a \iff q > h_\mu^{-1}\left(\frac{\gamma}{1-\gamma}\right)$$

where $h_\mu(q) := \frac{\mu}{1-\mu} \cdot h(\mu, q)$ and $h_\mu^{-1}(\cdot)$ is the inverse of $h_\mu(\cdot)$. In other words, for fixed $\mu > \frac{1}{2}$, we require a minimum fraction of honest workers to ensure that adversaries receive a higher penalty than honest workers, and this fraction increases as γ increases.

The above result shows that as the adversary strategies become smarter, it becomes more difficult to distinguish them from honest workers. Specifically, because $h_\mu^{-1}(\cdot)$ is a strictly increasing function, we require honest workers to have a larger majority as γ increases to ensure that they receive lower expected penalties than adversaries.

4.1.2 Asymptotic identification of adversaries and honest workers

Assuming that the expected penalties of adversaries and honest workers are separated, we now derive the asymptotic error rates, defined as the expected fraction of errors, of the soft-penalty algorithm as $n \to \infty$ when (1) there are no adversaries and (2) adversaries adopt the Uniform strategy.

To analyze the error rates, we consider the following threshold-classifier $\hat{\boldsymbol{I}}_\theta(\cdot)$ based on the soft-penalty algorithm: given a penalty threshold $\theta \in \mathbb{R}$, define the binary classifier

$$\hat{\boldsymbol{I}}_\theta(w_i) = \begin{cases} \text{honest}, & \text{if } \mathbf{PEN}_i \le \theta \\ \text{adversarial}, & \text{o.w.} \end{cases}$$

Let $\boldsymbol{I}(w_i)$ denote the latent true identity of worker w_i. Note that both $\boldsymbol{I}(w_i)$ and $\hat{\boldsymbol{I}}_\theta(w_i)$ are random variables under our generative model.

As above, we first consider the case in which there are no adversaries, that is, $q = 1$, so that $\boldsymbol{I}(w_i) = \textit{honest}$ for all workers w_i. In this case, Theorem 3 shows that workers with

higher reliabilities receive lower expected penalties. Furthermore, we now show that the threshold classifier correctly classifies high-reliability workers as honest with high probability as $n \to \infty$:

Theorem 6 (Identification of high-reliability workers) *Suppose $q = 1$ and $\mu > \frac{1}{2}$. Given $\theta \in (0,1)$ and $\varepsilon \in (0, \frac{1}{\sqrt{2}})$ such that $\theta - \varepsilon \in (g(1-\mu), g(\mu))$, define $\hat{\mu}(\theta) := \frac{g(\mu)+\varepsilon-\theta}{g(\mu)-g(1-\mu)}$, where the function $g(\cdot)$ was defined in Theorem 4. Then, under the generative model, we obtain*

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) \leq \frac{l^2 r^2}{n-1} + \exp\left(-\frac{2l\varepsilon^2}{(1-1/r)^2}\right)$$

When $l = \log n$ and r is fixed, we obtain

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right) \text{ as } n \to \infty$$

Theorem 6 provides an upper bound for the *error rate* or *misclassification rate* of the threshold classifier. We say that the classifier makes an *error* if it classifies a worker whose reliability is higher than $\hat{\mu}(\theta)$ as adversarial, and the misclassification rate is defined as the expected fraction of errors. As $n \to \infty$, the first term, $l^2 r^2/(n-1)$, in the error bound goes to 0 as long as $l = o(\sqrt{n})$. With the task degree r fixed, the second term goes to zero as long as $l \to \infty$ when $n \to \infty$. Upon combining these two observations, it follows that taking $l = \log n$ yields the error bound of $O(1/n^{2\varepsilon^2})$ for a fixed $\varepsilon \in (0, \frac{1}{\sqrt{2}})$ and r, as $n \to \infty$. In other words, our result shows that as long as we collect $\log n$ labels from each worker and a fixed number of labels for each task, we will classify workers with reliabilities higher than $\hat{\mu}(\theta)$ as honest with a high probability as $n \to \infty$. Although the number of labels collected from each worker must tend to infinity, it must only grow logarithmically in the total number of workers n. Finally, if the population reliability μ is known, we can determine the value of threshold θ for a given reliability threshold $\hat{\mu}(\theta)$.

The proof of Theorem 6 relies on establishing that the worker penalties are concentrated around their respective expectations, for which we need the worker-task assignment graph $\mathcal{B}$ to be locally tree-like:

Definition 7 (Locally tree-like assignment graphs) *An (l,r)-regular worker-task assignment graph $\mathcal{B}$ is said to be D-locally tree-like at a worker node w_i if the subgraph $\mathcal{B}_{w_i,D}$, consisting of nodes at a distance of at most D from w_i, is a tree.*

For our purposes, it suffices to have $\mathcal{B}_{w_i,2}$ be a tree. Note that the subgraph $\mathcal{B}_{w_i,2}$ consists of the worker node w_i; tasks labeled by w_i, that is, the set $\mathcal{T}_i$; and workers $\bigcup_{j \in \mathcal{T}_i} W_j$ who labeled the tasks in $\mathcal{T}_i$. Karger et al. (2014) show that a random construction of the assignment graph using the configuration model ensures that $\mathcal{B}_{w,2}$ is a tree with a high probability as $n \to \infty$ for a randomly chosen worker w.

Lemma 8 (Random construction ensures local tree-structure) *If $\mathcal{B}$ is random (l,r)-regular constructed according to the configuration model, then for a randomly chosen worker $\boldsymbol{w}$*

$$\Pr\left(\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree}\right) \leq \frac{l^2 r^2}{n-1}$$

The proof of the above lemma is in Appendix A.2.1 and it follows the arguments in Karger et al. (2014). Based on the result of Lemma 8, the proof of Theorem 6 proceeds in two steps. First, whenever the configuration model generates an assignment graph $\mathcal{B}$ that is not locally tree-like, we immediately declare an error, incurring an error probability that is bounded above by $l^2r^2/(n-1)$; this yields the first term in the error bound. Second, when $\mathcal{B}_{w_i,2}$ is indeed a tree, we obtain the second term in the error bound by invoking the following concentration result:

Lemma 9 (Concentration of honest worker penalties) *Suppose that $q = 1$ and $\mathcal{B}_{w_i,2}$ is a tree. Under the generative model and for a fixed reliability $\boldsymbol{M}_i = \mu_i$, given any $\varepsilon > 0$, the penalty assigned to honest worker w_i concentrates as*

$$\Pr\left(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \varepsilon \,\middle|\, \boldsymbol{M}_i = \mu_i\right) \leq \exp\left(\frac{-2l\epsilon^2}{(1-1/r)^2}\right)$$

Note that the above lemma holds for *any* fixed value of reliability μ_i. The proof of the above result relies on expressing the penalty scores as an average of l random variables and then invoking Hoeffding's concentration bound. The local tree-like property of the assignment graph $\mathcal{B}$ at worker node w_i ensures that the l random variables are mutually independent (which is required for Hoeffding's inequality).

Next, we consider the case in which there is a fraction $1-q$ of workers who are adversaries and adopt the Uniform strategy. Theorem 4 above provides the necessary and sufficient conditions for an honest worker to receive a lower expected penalty than an adversary, that is, for $p_h < p_a$. Under these conditions, we obtain the following result:

Theorem 10 (Identification of honest and adversarial workers) *Suppose $p_h < p_a$ and let $\theta \in (p_h + \varepsilon, p_a - \varepsilon)$ for some ε small enough such that $0 < \epsilon < (p_a - p_h)/2$. Then, under the generative model we obtain*

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{honest}\right) \leq \frac{l^2r^2}{n-1} + \exp\left(-\frac{2l\varepsilon^2}{(1-1/r)^2}\right) + F\left(\hat{\mu}(q,\theta)\right)$$

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{adversarial}\right) \leq \frac{l^2r^2}{n-1} + \exp\left(-\frac{2l\varepsilon^2}{(1-1/r)^2}\right)$$

where $\hat{\mu}(q,\theta)$ is such that $\hat{\mu}(1,\theta) = \hat{\mu}(\theta)$ and $\hat{\mu}(q,\theta) < \mu$ for all $q \in (0,1]$ and $\theta \in (p_h + \varepsilon, p_a - \varepsilon)$.

When $l = \log n$ and r is fixed, we obtain

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{honest}\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right) + F\left(\hat{\mu}(q,\theta)\right)$$

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{adversarial}\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right)$$

The precise expression for $\hat{\mu}(q,\theta)$ is involved and is given in Appendix A.2.4. Theorem 10 provides the misclassification rate of our algorithm when the population parameters q, μ, and γ satisfy the conditions of Theorem 4, ensuring that honest workers receive a lower expected penalty than adversaries. Following the arguments from the discussion under Theorem 6 above, it can be seen that when $l = \log n$ and r is fixed, the fraction of adversaries that is misclassified is $O(1/n^{2\varepsilon^2})$. On the other hand, the fraction of honest workers that is misclassified scales as $O(1/n^{2\varepsilon^2}) + F(\hat{\mu}(q,\theta))$. The first term tends to zero as $n \to \infty$. The second term denotes the probability that honest worker reliability is less than or equal to $\hat{\mu}(q,\theta)$. In other words, our algorithm misclassifies low-reliability workers as adversaries. In the special case when all honest workers have the same reliability μ, it immediately follows that the probability density function is a point mass at μ, from which it follows that $F(\hat{\mu}(q,\theta)) = 0$ because $\hat{\mu}(q,\theta) < \mu$. In this case, the misclassification error for the honest workers also tends to zero as $n \to \infty$.

We note that the dependence of the honest worker misclassification rate on $F\left(\hat{\mu}(q,\theta)\right)$ is fundamental to our algorithm. As an example, consider the case when the reliability distribution is a two-point distribution with probability mass μ at 1 and the remaining $1-\mu$ mass at 0. This distribution results in two types of honest workers: workers who always provide the correct response and those that always provide the incorrect response. Note that the average reliability under this distribution is μ. Let p_0 and p_1 denote the expected penalties under our generative model for a worker conditioned on being honest with reliabilities 0 and 1, respectively. Then, it follows from Lemma 19 in the Appendix that

$$\begin{aligned} p_1 &= \gamma \cdot g(1-P) + (1-\gamma) \cdot g(Q) \\ p_a &= \gamma \cdot g(1-P) + (1-\gamma) \cdot g(1-Q) \\ p_0 &= \gamma \cdot g(P) + (1-\gamma) \cdot g(1-Q) \end{aligned}$$

From Theorem 4, it follows that $P > 1/2$ and $Q < 1/2$ is necessary to ensure $p_h < p_a$. Combined with the fact that $g(\cdot)$ is increasing, this implies that $g(Q) < g(1-Q)$ and $g(1-P) < g(P)$. As a result, we obtain $p_1 < p_a < p_0$. It now follows that when $n \to \infty$, the penalties of honest workers with reliability 0 concentrate around p_0, and consequently, they are classified as adversarial whenever the threshold $\theta < p_a$, resulting in a misclassification error of $1-\mu$. However, it should be noted that honest workers classified as adversaries indeed have low reliabilities.

Similar to the proof of Theorem 6 above, the proof of Theorem 10 proceeds in two steps. The first term in the error bound of Theorem 10 comes from Lemma 8 because we immediately declare an error whenever the assignment graph is not locally tree-like. The second term comes from the case when $\mathcal{B}_{w_i,2}$ is indeed a tree by invoking the following concentration result:

Lemma 11 (Concentration of worker penalties) *Suppose that $q < 1$ and $\mathcal{B}_{w_i,2}$ is a tree. Under the generative model, given any reliability value $\hat{\mu} \in (0,1)$ and $\varepsilon > 0$, the penalty assigned to worker w_i concentrates as*

$$\Pr\left(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \hat{\mu}] + \varepsilon \;\middle|\; \boldsymbol{I}(w_i) = \textit{honest}\right) \leq \exp\left(\frac{-2l\epsilon^2}{(1-1/r)^2}\right) + F(\hat{\mu})$$

and

$$\Pr\left(\mathbf{PEN}_i \leq p_a - \varepsilon \;\middle|\; \boldsymbol{I}(w_i) = \textit{adversarial}\right) \leq \exp\left(\frac{-2l\epsilon^2}{(1-1/r)^2}\right)$$

The proof of the above result is similar to that of Lemma 9. For the case of adversarial workers, we use Hoeffding's argument to establish the concentration. For the case of honest workers, the first term follows directly from Lemma 9 when w_i has reliability $\mu_i > \hat{\mu}$ and the second term is the probability that the reliability $\mu_i \leq \hat{\mu}$.

The above results establish that the soft-penalty algorithm successfully identifies low-reliability honest workers and adversaries adopting the Uniform strategy asymptotically with high probability. Note that all results also extend to the random, normalized variant of the hard-penalty algorithm mentioned in Section 3.3, where the expectation is taken over the generative model *and* the randomized hard-penalty algorithm (see Appendix A.3).

We would like to note that similar results can be derived if it is assumed that honest workers employ the two-coin model (instead of the one-coin model assumed in the preceding analysis). However, the precise error bounds require significantly more notation to explain, without adding too much in terms of insights. Instead, we evaluate the performance of our algorithm for the two-coin model in the numerical experiments and show that it is still able to identify uniform adversaries and low-reliability honest workers (see Section 5.1).

4.2 Hard-penalty algorithm: sophisticated adversary strategies

In the preceding analysis, we focused on common adversary strategies in which adversaries were not intentionally malicious. However, existing studies provide ample evidence for the presence of workers with malicious intent in public crowdsourcing systems, where workers choose which tasks to label and the worker labels are public. These workers are usually hired online by an attacker (Wang et al., 2012) to create fake accounts and manipulate ratings/reviews to alter the aggregate ratings or rankings received by tasks. Specific examples include workers on Digg altering the "popularity" of advertisements and phishing articles (Tran et al., 2009), fake review groups collaboratively targeting products on Amazon (Mukherjee et al., 2012), workers providing fake ratings and reviews to alter the aggregate ratings of restaurants on Yelp (Molavi Kakhki et al., 2013), and malicious crowd behavior in online surveys (Gadiraju et al., 2015). Refer to the recent work by Wang et al. (2014) for more examples. Motivated by these examples, we study settings with *sophisticated adversaries*, who are defined as follows:

Definition 12 (Sophisticated adversaries) *Sophisticated adversaries provide responses with the objective of maximizing the number of tasks whose inferred labels are different from the labels they would otherwise have received from any label aggregation algorithm. They are computationally unbounded and colluding, and they possess knowledge about the labels provided by honest workers; therefore, they can adopt arbitrary response strategies.*

Our definition allows sophisticated adversaries to not just be malicious but also be capable of executing the most complex strategies. In practice, an adversary may adopt feasible strategies with varying complexities depending on the application context and their objectives. Focusing on the most sophisticated adversary makes our analysis broadly applicable,

independent of the application context. Further, we do not restrict the structure of the worker-task assignment graph because unlike in a crowdsourced-classification task, we have no control on which tasks each worker labels.

We first note that existing label aggregation algorithms can have arbitrarily bad performance in the presence of sophisticated adversaries, even if we assume that all honest workers are perfectly reliable:

Lemma 13 (Hardness of recovering true task labels) *Suppose honest workers are perfectly reliable, that is, they always provide the correct response. Let the assignment graph between honest workers and tasks be such that each task receives responses from at most $r > 1$ honest workers. Suppose there are $k > r$ sophisticated adversaries, and each adversary provides the incorrect label on all tasks. Then, both the simple majority and EM (initialized by majority estimates) label aggregation algorithms output the incorrect label for all tasks.*

Note that the above lemma characterizes the performance of label aggregation algorithms, which are designed to infer true task labels as opposed to identifying adversarial workers. Because sophisticated adversaries aim to maximize the number of tasks whose inferred labels are incorrect, the lemma shows that for standard label aggregation algorithms like simple majority and EM, the adversaries can cause arbitrarily bad "damage". Consequently, in our theoretical analysis below, we focus on the accuracy of label aggregation algorithms in the presence of sophisticated adversaries, as opposed to the misclassification rate, as was done in Section 4.1.

Lemma 13 holds for r-right regular graphs, that is, each task receives exactly r honest worker labels, which are commonly used in practice (see the discussion in Section 4.1). Therefore, standard label aggregation algorithms can have very poor accuracy in recovering true task labels in the presence of sophisticated adversaries. In fact, we can actually establish something much stronger—in the presence of sophisticated adversaries, there exists a lower bound on the number of tasks whose true label cannot be inferred correctly (better than random guess) *irrespective* of the label aggregation algorithm used to aggregate the worker responses.

4.2.1 Lower bound on number of tasks that receive incorrect labels

To state our result, we need the following additional notation. We represent any label aggregation algorithm as a *decision rule* $\boldsymbol{R} : \mathcal{L} \to \{-1, +1\}^m$, which maps the observed labeling matrix $\mathcal{L}$ to a set of output labels for each task. Because of the absence of any auxiliary information about the workers or the tasks, the class of decision rules, say $\mathcal{C}$, is invariant to permutations of the identities of workers and/or tasks. More precisely, $\mathcal{C}$ denotes the class of decision rules that satisfy $\boldsymbol{R}(P\mathcal{L}Q) = \boldsymbol{R}(\mathcal{L})Q$ for any $n \times n$ permutation matrix P and $m \times m$ permutation matrix Q. We say that a task is *affected* if a decision rule outputs the incorrect label for the task, and we define the *quality* of a decision rule $\boldsymbol{R}(\cdot)$ as the *worst-case* number of affected tasks over all possible true labelings of the tasks and adversary strategies given a *fixed* set of honest worker responses. Fixing the responses provided by honest workers allows isolation of the effect of the adversary strategy on the accuracy of the decision rule. Considering the worst-case over all possible true task labelings makes the quality metric robust to ground-truth assignments, which are typically application specific.

To formally define the quality, let $\mathcal{B}_H$ denote the subgraph of the worker-task assignment graph restricted to honest workers H and $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ denote the vector of true labels for the tasks. Because the number of affected tasks depends on the actual honest worker responses, we focus on the case when all the honest workers are perfectly reliable, that is, they always provide the correct response. Focusing on completely reliable honest workers allows us to isolate the impact of adversaries because any misidentification is caused by the presence of adversaries. Finally, let $\mathcal{S}_k$ denote the strategy space of $k < |H|$ adversaries, where each strategy $\sigma \in \mathcal{S}_k$ specifies the $k \times m$ response matrix given by the adversaries. Because we do not restrict the adversary strategy in any way, it follows that $\mathcal{S}_k = \{-1, 0, +1\}^{k \times m}$. The *quality* of a decision rule $\boldsymbol{R} \in \mathcal{C}$ is then defined as

$$\mathrm{Aff}(\boldsymbol{R}, \mathcal{B}_H, k) = \max_{\sigma \in \mathcal{S}_k, \boldsymbol{y} \in \{-1,+1\}^m} \left| \left\{ t_j \in \mathcal{T} \colon R_{t_j}^{\boldsymbol{y},\sigma} \neq y_j) \right\} \right|$$

where $R_t^{\boldsymbol{y},\sigma} \in \{-1, +1\}$ is the label output by the decision rule $\boldsymbol{R}$ for task t when the true label vector is $\boldsymbol{y}$ and the adversary strategy is σ. Note that our notation $\mathrm{Aff}(\boldsymbol{R}, \mathcal{B}_H, k)$ makes the dependence of the quality measure on the honest worker subgraph $\mathcal{B}_H$ and the number of adversaries k explicit.

We obtain the following result, which establishes a lower bound on the quality of any decision rule:

Theorem 14 (Lower bound on number of affected tasks) *Suppose that $|A| = k$ and all honest workers provide correct responses. Let $\mathrm{PreIm}(\mathcal{T}')$ denote the set of honest workers who label at least one task in $\mathcal{T}' \subseteq \mathcal{T}$. Then, given any honest worker-task assignment graph $\mathcal{B}_H$, there exists an adversary strategy $\sigma^* \in \mathcal{S}_k$ that is independent of any decision rule $\boldsymbol{R} \in \mathcal{C}$, such that*

$$\max_{\boldsymbol{y} \in \{-1,+1\}^m} \mathrm{Aff}(\boldsymbol{R}, \sigma^*, \boldsymbol{y}) \geq L \quad \forall \boldsymbol{R} \in \mathcal{C}, \quad \textit{where}$$

$$L := \frac{1}{2} \cdot \left(\max_{\substack{\mathcal{T}' \subseteq \mathcal{T}: \\ |\mathrm{PreIm}(\mathcal{T}')| \leq k}} |\mathcal{T}'| \right)$$

and $\mathrm{Aff}(\boldsymbol{R}, \sigma^*, \boldsymbol{y})$ denotes the number of affected tasks under adversary strategy σ^*, decision rule $\boldsymbol{R}$, and true label vector $\boldsymbol{y}$ (with the assumption that the maximum over an empty set is zero). In particular, this means that $\mathrm{Aff}(\boldsymbol{R}, \mathcal{B}_H, k) \geq L$ for all decision rules $\boldsymbol{R} \in \mathcal{C}$. In addition, there exist honest worker-task assignment graphs such that σ^* renders the adversaries A indistinguishable from a set of k honest workers, so that a random guess misclassifies $2k$ workers with probability $(1/2)$.

We describe the main idea of the proof. The proof proceeds in two steps: (i) we provide an explicit construction of adversary strategy σ^* that depends only on $\mathcal{B}_H$ and (ii) we show the existence of two possible true labelings $\tilde{\boldsymbol{y}} \neq \boldsymbol{y}$ such that $\boldsymbol{R}$ outputs exactly the same labels in both scenarios. The adversary labeling strategy we construct uses the idea of *indistinguishability*, which captures the fact that by carefully choosing their responses, adversaries can render themselves indistinguishable from honest workers. In the simple case

when there is only one honest worker, the adversary simply flips the response provided by the honest worker, so that each task will have two labels of opposite parity. It can be argued that because there is no other discriminatory information, it is impossible for any decision rule $\boldsymbol{R} \in \mathcal{C}$ to distinguish the honest worker from the adversary and to therefore identify the true label of any task (better than a random guess). We extend this to the general case, where the adversary "targets" at most k honest workers and derives a strategy based on the subgraph of $\mathcal{B}_H$ restricted to the targeted workers. The resultant strategy can be shown to result in incorrectly identified labels for at least L tasks for some ground-truth label assignment.

Note that Theorem 14 holds for *any* honest worker-task assignment graph $\mathcal{B}_H$. This is particularly remarkable given that the analysis of aggregation algorithms becomes extremely complicated for general graphs (a fact observed in previous studies; see Dalvi et al., 2013). The bound L itself depends on the structure of $\mathcal{B}_H$, and therefore, it can be difficult to interpret in general. However, it becomes interpretable for (r, γ, α)-bipartite expanders, as defined next.

Definition 15 (Expanders) *An honest worker-task assignment graph $\mathcal{B}_H = (H \cup \mathcal{T}; E)$, with edges E between the honest workers H and tasks $\mathcal{T}$, is (r, γ, α)-bipartite expander if (i) $\mathcal{B}_H$ is r-right regular, that is, each task is labeled by r honest workers and (ii) for all $\mathcal{T}' \subseteq \mathcal{T}$ such that $|\mathcal{T}'| \leq \gamma |\mathcal{T}|$, the pre-image of $\mathcal{T}'$ satisfies $|\text{PreIm}(\mathcal{T}')| \geq \alpha |\mathcal{T}'|$, where $\text{PreIm}(\mathcal{T}')$ is the set of all honest workers who label at least one task in $\mathcal{T}'$.*

Note that the definition entails that $\alpha \leq r$. We have the following corollary of Theorem 14 when $\mathcal{B}_H$ is (r, γ, α)-bipartite expander.

Corollary 16 (Lower bound for expanders) *Suppose $\mathcal{B}_H$ is (r, γ, α)-bipartite expander. Then, k adversary identities can affect at least L tasks such that $\lfloor \frac{k}{r} \rfloor \leq 2L \leq \lceil \frac{k}{\alpha} \rceil$, provided $\lceil \frac{k}{\alpha} \rceil + 1 < \gamma \cdot |\mathcal{T}|$. Furthermore, given any constant r, there exists $\gamma > 0$ such that a uniformly random $\mathcal{B}_H$ is $(r, \gamma, r-2)$-bipartite expander with probability at least $1/2$, in which case the lower bound $L = \frac{1}{2} \lceil \frac{k}{r-2} \rceil$.*

The proof is provided in Appendix A.5. The above statement says that if the honest worker-task assignment graph $\mathcal{B}_H$ is constructed randomly, then k adversary identities can affect at least $\frac{1}{2} \lfloor \frac{k}{r} \rfloor$ tasks. The bound implies that the ability of the adversaries to affect tasks increases linearly as the number of identities k increases. Further, the damage that k adversaries can do decreases inversely with the number of honest workers r who provide labels for each task. Both implications are intuitive. As can be seen from the proof, the lower bound $\frac{1}{2} \lfloor \frac{k}{r} \rfloor$ on L in Corollary 16 holds for all r-right regular graphs, even if they are not expanders.

Having established the above lower bound and in light of Lemma 13, the natural question to ask is as follows: does there exist a label aggregation algorithm for which we can prove an upper bound on the number of affected tasks, irrespective of the strategy employed by the sophisticated adversaries? Below, we show such a label aggregation algorithm that is a natural extension of the hard-penalty algorithm.

4.2.2 Accuracy of hard-penalty algorithm

We introduce the *penalty-based label aggregation* algorithm (see Algorithm 3) for our analysis, which is a natural extension of the hard-penalty algorithm to also perform label aggregation:

Algorithm 3 penalty-based label aggregation

1: **Input:** W, $\mathcal{T}$, and $\mathcal{L}$
2: Perform steps 2 and 3 of the hard-penalty algorithm
3: For each task t_j, let $w_{t_j^+}, w_{t_j^-}$ be worker nodes that task nodes t_j^+, t_j^- are respectively mapped to in optimal semi-matching $\mathcal{M}$ in Step 2
4: **Output**

$$\hat{y}_j = \begin{cases} +1 & \text{if } deg_{\mathcal{M}}(w_{t_j^+}) < deg_{\mathcal{M}}(w_{t_j^-}) \\ -1 & \text{if } deg_{\mathcal{M}}(w_{t_j^+}) > deg_{\mathcal{M}}(w_{t_j^-}) \\ \leftarrow \{-1, +1\} & \text{otherwise} \end{cases}$$

(here $\hat{y}_j$ refers to the output label for task t_j and $\leftarrow \{-1, +1\}$ means that $\hat{y}_j$ is drawn uniformly at random from $\{-1, +1\}$)

For our analysis, we consider the following model for honest worker responses that is based on the *spammer-hammer* model popularly used in previous studies (Karger et al., 2011; Liu et al., 2012; Ok et al., 2016; Khetan and Oh, 2016): $0 \leq \varepsilon < 1$ fraction of honest workers are "spammers," that is, they make mistakes in their responses, and the remaining $1 - \varepsilon$ fraction are "hammers," that is, they are perfectly reliable and always provide the correct response. In the theorem below, $\mathcal{B}_H^{cs}$ refers to the bipartite graph created as follows: (i) each honest worker h is represented by a node on the left; (ii) each task $t_j \in \mathcal{T}$ is represented by (at most) two nodes on the right, t_j^+ and t_j^-; and (iii) add the edge (h, t_j^+) (resp. (h, t_j^-)) if honest worker h labels task t_j as $+1$ (resp. -1). Then, we can show the following:

Theorem 17 (Accuracy of penalty-based label aggregation) *Suppose that $|A| = k$ and let $\varepsilon \in [0, 1)$ denote the fraction of honest workers who make mistakes in their responses. Furthermore, let $d_1 \geq d_2 \geq \cdots \geq d_{|H|}$ denote the degrees of honest workers in the optimal semi-matching on $\mathcal{B}_H^{cs}$. For any true labeling $\boldsymbol{y}$ of the tasks and under Algorithm 3 (with the convention that $d_i = 0$ for $i > |H|$):*

1. *If $\varepsilon = 0$, there exists an adversary strategy σ^* such that the number of affected tasks is at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$*
2. *Assuming that each task receives at least one correct response from the honest workers, no adversary strategy can affect more than U tasks where*
   (a) $U = \sum_{i=1}^{k+\varepsilon\cdot|H|} d_i$, *when at most one adversary provides correct responses*
   (b) $U = \sum_{i=1}^{2k+\varepsilon\cdot|H|} d_i$, *in the general case*

A few remarks are in order. First, it can be shown that for optimal semi-matchings, the degree sequence $d_1, d_2, \ldots, d_{|H|}$ is unique (see the proof in Appendix A.7), and therefore, the bounds in the theorem above are uniquely defined given $\mathcal{B}_H^{cs}$. The result of Theorem 17 provides both a lower and upper bound for the number of tasks that can be affected by k adversaries under the penalty-based label aggregation algorithm, irrespective of the adversary strategy. This is remarkable, especially because we established that existing label aggregation algorithms can be arbitrarily bad (Lemma 13). Assuming honest workers are always correct, that is, $\varepsilon = 0$, our characterization is reasonably tight when all but (at most) one adversary provide incorrect responses. In this case, the gap between the upper and a constant factor of the lower bound is d_k, which can be "small" for large enough k. However, our characterization is loose in the general case when adversaries can provide arbitrary responses. Here, the gap is $\sum_{i=k}^{2k} d_i$; we attribute this to our proof technique and conjecture that the upper bound of $\sum_{i=1}^{k} d_i$ also applies to the more general case. When $\varepsilon > 0$, the upper bound U increases because there are more incorrect responses and, in turn, the scope for adversaries to affect a larger number of tasks.

One might wonder whether we could perform a similar analysis for the case when honest workers follow the one-coin model. Given the complexity of analyzing optimal semi-matchings, our current proof technique does not readily extend to this scenario, and therefore, there is an opportunity to improve the analysis in future work.

Optimality of penalty-based label aggregation. We now compare the upper bound U in Theorem 17 to the lower bound L in Theorem 14 in the case when honest workers are perfectly reliable, that is, $\varepsilon = 0$. We show that (see Appendix A.8) when the degrees $d_1, d_2, \ldots, d_{|H|}$ are all distinct, $L \geq \frac{1}{2}\sum_{i=1}^{k-1} d_i$. Combined with Theorem 14, this shows that k adversaries can affect at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks *irrespective* of the label aggregation algorithm used to aggregate the worker responses. From Theorem 17, we also have that under the penalty-based label aggregation algorithm, k adversaries can affect at most $U = \sum_{i=1}^{2k} d_i \leq 3(\sum_{i=1}^{k-1} d_i)$ tasks (as long as $k \geq 2$). Therefore, our algorithm achieves constant factor optimality in recovering the true labels of tasks *irrespective* of the adversary's strategy.

5. Numerical analysis

We conducted two empirical studies to demonstrate the practical value of our methods. The first study illustrates a concrete real-world application of our methodology. By using standard crowdsourcing datasets (as described below), it shows that filtering out the adversaries identified by our methods allows existing label aggregation algorithms to infer true task labels more accurately. Such improvements in accuracy by *discarding* labels from certain workers suggests that their label patterns do not conform to standard probabilistic assumptions. Although not illustrated in our study, instead of being filtered out, the adversary labels may also be used by fitting a model different from that of honest workers in applications where such assumptions are reasonable. The second study is designed to assess the ability of our methods to successfully identify adversaries and low-reliability honest workers. It is a simulation study in which we injected a standard crowdsourcing task with "spammers" (workers who label a task $+1$ and -1 with probability $1/2$ each irrespective of the true label) and workers adopting the Uniform strategy. It demonstrates that both soft- and hard-penalty algorithms successfully identify adversaries and low-reliability

honest workers when the worker-task assignment graph has a power-law degree distribution for workers and tasks, thus complementing the results in Section 4.1 which focused on (l, r)-regular worker-task assignment graphs.

For the purposes of our studies, we focused on the following six label aggregation algorithms: (a) simple majority algorithm MV; (b) EM algorithm for the two-coin model (Raykar and Yu, 2012); (c) KOS algorithm (Karger et al., 2014); (d) KOS(NORM), a normalized variant of KOS in which messages are scaled by the corresponding worker and task degrees to account for non-regular node degrees in the assignment graph (see Appendix B for the precise description); (e) SPEC-EM, the spectral EM algorithm of Zhang et al. (2014); and (f) MMCE, the regularized minimax conditional entropy approach of Zhou et al. (2015). We implemented both variants of our reputation algorithm: (a) soft-penalty (SOFT) and (b) hard-penalty (HARD). As removing workers alters the penalties of the remaining workers, we filtered the workers iteratively. In each iteration, we recomputed the penalties of the remaining workers, removed the worker with the highest penalty, and repeated until a prespecified number of workers was removed. [7]

Finally, as mentioned in Section 4.1, we implemented two variants of the soft-penalty algorithm: one in which non-conflict tasks are dropped for assigning worker penalties and another in which they are retained. The results were essentially the same, so we only report the results for the variant in which the non-conflict tasks were dropped.

Dataset	Workers	Tasks	Responses
rte	164	800	8000
temp	76	462	4620
stage2	68	711	2035
task2	386	2269	12435
tweets	66	1000	4977

Table 1: Summary of real datasets used in the experiments

5.1 Accuracy improvements on real-world crowdsourcing datasets

We focused on the following standard datasets:

- `stage2` and `task2`: consisting of a collection of topic-document pairs labeled as relevant or non-relevant by workers on Amazon Mechanical Turk (see Tang and Lease, 2011). These datasets were collected as part of the TREC 2011 crowdsourcing track.
- `rte` and `temp`: consisting of annotations by Amazon Mechanical Turk workers for different natural language processing (NLP) tasks. `rte` consists of binary judgments for textual entailment (whether one sentence can be inferred from another) and `temp` for temporal ordering of events (see Snow et al., 2008).
- `tweets`: consisting of sentiment (positive or negative) labels for 1000 tweets (see Mozafari et al., 2014).

7. The performance was similar for the non-iterative variant discussed in Section 3—we report the results for the iterative version because it had marginally better performance.

As inferring the reliabilities of workers who labeled very few tasks [8] is difficult, we preprocessed the datasets to remove all workers who labeled less than three tasks. Table 1 shows summary statistics of the datasets after our preprocessing.

Table 2 reports the accuracies of various label aggregation algorithms for inferring true task labels. For each benchmark label aggregation algorithm, the column BASE reports the accuracy of the algorithm in isolation. The columns SOFT and HARD report the best accuracy of the algorithms for filtering $k = 1, 2, \ldots, 10$ workers using the soft- and hard-penalty algorithms, respectively; the numbers in parentheses indicate the k value for which we observed the best performance. [9]

Algorithm	rte			temp			stage2			task2			tweets		
	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD
MV	91.9	92.1(7)	**92.5(3)**	93.9	93.9	**94.4(5)**	74.3	75.4(1)	**80.5(2)*****	64.2	64.2	**67.8(10)*****	69.6	69.8(4)	**73.3(1)*****
EM	93.0	93.0	**93.3(5)**	94.1	94.1	94.1	70.2	76.8(4)	**81.2(6)*****	67.0	67.1(6)	**68.6(9)*****	71.2	71.2	**71.7(1)**
KOS	49.7	89.0(10)	**91.6(10)*****	56.9	69.3(4)	**93.7(3)*****	74.5	74.7(7)	**75.1(2)**	57.4	57.4	**65.6(10)*****	65.8	66.0(4)	**70.5(1)*****
KOS(NORM)	91.3	92.6(5)	**93.1(6)****	93.9	94.4(7)	**94.4(1)**	75.5	75.5	**78.2(2)***	58.3	58.9(8)	**67.7(9)*****	68.7	68.7	**71.0(2)*****
SPEC-EM	90.1	92.4(8)	**92.8(6)**	56.1	94.2(7)	**94.4(5)**	82.8	82.8	82.8	56.1	56.1	**57.6(10)*****	67.0	67.0	**69.0(1)*****
MMCE	92.5	92.9(9)	**93.2(3)**	94.4	94.4	94.4	57.0	58.5(5)	**73.3(10)*****	66.1	66.1	**66.9(10)**	72.7	72.7	72.7

Table 2: **Percentage accuracy** in recovering true labels of benchmark algorithms in isolation and when combined with our modified reputation algorithms. For each benchmark, the best-performing combination is highlighted in bold. The number in parentheses represents the number of workers filtered by our reputation algorithm (an absence indicates that no performance improvement was achieved while removing upto 10 workers with the highest penalties). The p-values, according to a two-sided paired t-test, are denoted as * $p < 0.1$, ** $p < 0.05$, and *** $p < 0.01$.

The key conclusion we draw is that filtering out workers flagged by our algorithms as adversaries boosts the predictive accuracy of state-of-the-art aggregation algorithms significantly across the datasets: the average improvement in accuracy for MV is 3.7%, EM is 3.4%, KOS is 30.2%, KOS(NORM) is 4.4%, SPEC-EM is 12.6%, and MMCE is 4.7% when using the hard-penalty algorithm. The improvement is large for KOS because it is designed for regular graphs (with all workers having the same degree and all tasks having the same degree) and suffers in performance on real-world graphs that are not regular. Second, we note that our methods can boost the performance of the MV and KOS algorithms to the level of the popular EM algorithm. The MV algorithm is simple to implement, and the KOS algorithm is designed for scenarios where the underlying assignment graph is random (l, r)-regular and has strong theoretical guarantees and robustness to different initializations (Karger et al., 2014). Our results suggest that implementing the MV and KOS algorithms in conjunction with our reputation algorithms can allow us to obtain their respective simplicity and robustness along with strong practical performance (even for irregular graphs) comparable

8. The datasets `stage2`, `task2`, and `tweets` contain several workers who provided responses for only a single task.
9. The performance was robust to the choice of k. We matched or improved the accuracy of the underlying label aggregation algorithm in 66% and 70% of cases on average for the soft- and hard-penalty algorithm, respectively.

to that of the EM algorithm. [10] Finally, because discarding the labels of certain workers improves the predictive accuracy, our results suggest that standard probabilistic models (including the two-coin model) are insufficient to capture the labeling patterns of workers in real-world datasets.

To gain insights into the types of workers identified by our algorithms, we conducted a qualitative analysis of the labeling patterns of the workers that were filtered out. We observed the following key types:

1. Workers who labeled at least 10 tasks, of which more than 95% had the same label. For instance, our algorithms detected six such workers in the `temp` dataset, five in the `task2` dataset, and one in the `tweets` dataset. For the `stage2` dataset, we detected two workers who gave all $+1$ labels and one worker who gave all but a single response as -1; it should be noted that the empirically calculated prevalence γ of $+1$ tasks in the `stage2` dataset was 0.83, potentially suggesting that the two workers who gave all $+1$ labels were adopting "smart" strategies.

2. Workers who provide labels independent of true task labels. For instance, we detected four such workers in the `rte` dataset, seven workers in the `temp` dataset, seven in the `task2` dataset, three in the `stage2` dataset, and one in the `tweets` dataset, whose label patterns are such that the empirical fractions $\hat{\alpha}$ and $\hat{\beta}$ of correct responses among the tasks with true labels $+1$ and -1, respectively, satisfy $|\hat{\alpha} + \hat{\beta} - 1| \leq 0.05$. Raykar and Yu (2012) showed that such workers effectively assign a label of $+1$ with probability $\hat{\alpha}$ and -1 with probability $1 - \hat{\alpha}$ independent of the true task label.

3. Workers with skewed reliabilities. Such workers were accurate on tasks with one type of true label, say, $+1$, but not on others, say, tasks with true label -1. Such label patterns of workers may be indicative of tasks that require subjective assessments. For instance, we found four workers in the `tweets` dataset and two workers in `stage2` dataset that had skewed reliabilities. In the `tweets` dataset, workers were asked to rate the sentiment of a tweet as being positive or negative. As the notion of tweet sentiment can be subjective, workers with biased views of the sentiment make systematic errors on one type of task. A similar explanation applies to the `stage2` dataset, in which workers were asked to label a topic-document pair as relevant or not, requiring a potentially subjective assessment. See Kamar et al. (2015) for more examples.

In summary, our reputation algorithms are successful in identifying adversarial workers adopting a broad set of strategies. Furthermore, although not reported, the empirical reliabilities of the workers filtered out using our algorithms were on average *lower* than those of unfiltered workers. This suggests that the labels our algorithms discard are from low-reliability workers who provide little to no information about the true task labels; this explains the improved accuracy we obtain.

10. Note that the EM algorithm can also have theoretical guarantees with appropriate initializations (see Gao and Zhou, 2016 and Zhang et al. 2014).

5.2 Identifying low-reliability honest workers and adversaries

We use a simulation study to show that our reputation algorithms successfully identify low-reliability honest workers and adversaries when the worker-task assignment graphs have power-law degree distributions for the worker and task nodes. Such graph structures are common in many real-world crowdsourcing scenarios (Franklin et al., 2011). The results of the simulation study complement the theoretical results presented in Section 4.1 for (l, r)-regular assignment graphs, which show that the soft-penalty algorithm successfully identifies low-reliability honest workers and adversaries adopting the Uniform strategy.

For our study, we used the following broad procedure: (a) generate a random crowdsourcing instance from the ground-truth model, (b) generate synthetic responses from the workers for a sample of tasks, (c) filter out workers with the highest penalties according to our reputation algorithms, and (d) compute the *precision*, that is, the fraction of filtered-out workers who are adversarial and who have low empirical reliabilities.

Setup of study. We considered a total of $n = 100$ workers. The probability q that a worker is honest was set to 0.7; therefore, on average, there are 30 adversaries among the 100 workers. The prevalence γ of $+1$ tasks was set to 0.5. We sampled worker degrees according to a power-law distribution (with exponent $a = 2.5$) with the minimum degree equal to 5, and then, we used the Python networkx library (Hagberg et al., 2008) to generate the worker-task assignment graph.[11] Note that the number of tasks m is determined from the total number of workers and the sampled worker degrees.

As the worker degrees are skewed, the performance of the algorithms is influenced by the adversary degrees. To capture this, we considered two scenarios: (a) adversaries have high degrees and (b) adversaries have low degrees. To ensure that adversaries on average have high degrees, we set worker w in the sampled worker-task assignment graph to be honest with probability $q_w = q \cdot (d_{\max} - d_w)/(d_{\max} - d_{\text{avg}})$ and to be adversarial with probability $1 - q_w$, where d_w is the degree of worker w and $d_{\max}$, $d_{\min}$, and d_{avg} are the maximum, minimum, and average degrees, respectively. Similarly, to ensure that adversaries on average have lower degrees, we set $q_w = q \cdot (d_w - d_{\min})/(d_{\text{avg}} - d_{\min})$. See Appendix B for details.

For each scenario, after the worker-task assignment graph and worker identities were sampled, we generated the crowdsourcing instances as follows: (a) set the true label y_j of each task t_j to be $+1$ with probability $1/2$ and -1 with probability $1/2$; (b) for each honest worker w, sample its reliability μ_w u.a.r from the interval $[0.8, 1.0)$, and set its response to each assigned task t_j to be the true label y_j with probability μ_w and $-y_j$ with probability $1-\mu_w$; and (c) generate responses from adversarial workers according to the chosen strategy. We focused on two adversary strategies: (a) Spammer—label each task $+1$ or -1 with prob. $1/2$ and (b) Uniform—label every assigned task $+1$. The first strategy reflects the setting of Theorem 3 because it is captured by the one-coin model, and the second strategy reflects the setting of Theorem 4.

Results. Table 3 reports the *precisions* of the SOFT and HARD algorithms when (iteratively) removing 10 workers. The precision is defined as the fraction of filtered-out workers who are either adversarial or honest with empirical reliabilities less than 0.85 (which is less

11. Specifically, we used the following function: `https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.bipartite.generators.preferential_attachment_graph.html`.

	Spammer		Uniform	
	Low Deg	High Deg	Low Deg	High Deg
SOFT	**90.03**	88.70	**77.13**	69.57
HARD	89.00	**93.83**	73.43	**76.73**

Table 3: The *precisions* of the soft- and hard-penalty algorithms in identifying low-reliability honest workers and adversaries when filtering 10 workers. The rows SOFT and HARD correspond to the soft- and hard-penalty algorithms, respectively. The columns Spammer and Uniform correspond to the two adversary strategies, and for each adversary strategy, the columns "Low Deg" and "High Deg" refer to the scenarios in which low- and high-degree workers, respectively, are more likely to be adversaries. Refer to the text for more details.

than the mean $\mu = 0.9$ of the honest worker reliability distribution), where the empirical reliability of an honest worker is equal to the fraction of its responses that were correct. The table reports the precision of the algorithms for the two adversary strategies, Spammer and Uniform, under the two scenarios: Low Deg, in which low-degree workers are more likely to be adversaries, and High Deg, in which high-degree workers are more likely to be adversaries. For each combination of adversary strategy and scenario, the reported numbers are the precision values averaged over 300 randomly generated instances.

We draw the following key conclusions. First, our algorithms have precision values $> 69\%$ in all scenarios, indicating that they are successful in identifying adversaries and low-reliability honest workers when the worker and task degrees have a power-law distribution. This finding complements the results of Theorem 6 and Theorem 10, which establish a similar result when the worker-task assignment graph is (l, r)-regular. Second, our results offer insights into the settings under which the soft- or hard-penalty algorithm is appropriate. Generally, we observe that the hard-penalty algorithm is more appropriate when the adversaries have higher degrees, whereas the soft-penalty algorithm is more appropriate when the adversaries have lower degrees. We also note that when the adversaries have labeling patterns (such as Spammer) that are probabilistically similar to those of honest workers, the soft-penalty algorithm has a performance comparable to that of the hard-penalty algorithm even when the adversaries have high degrees.

6. Conclusions

This study investigated the problem of identifying a broad class of adversarial workers in crowdsourced systems, when the population of workers consists of a mixture of honest workers and adversaries. Honest workers may be reliable or unreliable, and they provide labels according to a well-defined probabilistic model. The adversaries adopt strategies different from those of honest workers, whether probabilistic or not. Under this setting, we make algorithmic, theoretical, and empirical contributions. The key algorithmic contribution is the design of two reputation-based algorithms—soft-penalty and hard-penalty algorithms—that analyze workers' label patterns and assign a reputation score to each worker that indicate

the degree to which the labeling pattern is a statistical outlier. Under standard probabilistic assumptions, we show that the reputation scores assigned by the soft-penalty algorithm are consistent with worker reliabilities (probability that a worker provides the true task label) when there are no adversaries. We also show that under appropriate conditions, the soft-penalty algorithm can asymptotically separate adversaries from honest workers when adversaries adopt deterministic strategies. When adversaries are sophisticated, we derive a lower bound for the number of tasks that k adversaries can affect (true label cannot be inferred better than a random guess) and a corresponding upper bound for the number of affected tasks under the hard-penalty algorithm. Empirically, we show that our algorithm can be used to significantly enhance the accuracy of existing label aggregation algorithms in real-world crowdsourcing datasets.

To the best of our knowledge, our work is the first to consider general worker strategies in crowdsourced labeling tasks. Our work opens the doors for several exciting future directions. Both of our penalty-based algorithms assume that the task labels are binary; analyzing natural extensions of our algorithms to multi-class settings is an interesting direction. Because our algorithm allows the identification of adversarial workers, it could be combined with adaptive techniques to recruit more workers (Ramesh et al., 2012) and identify the right workers (Li et al., 2014). Finally, applying our outlier detection technique to ensemble learning approaches, in which outputs from multiple learning algorithms are combined, could be a promising future direction; indeed there has already been some work in this space (Wang and Yeung, 2014).

Acknowledgments

Ashwin Venkataraman is partially supported by NSF CAREER grant CMMI 1454310. We gratefully acknowledge the thoughtful and detailed feedback of the anonymous referees which led to a substantial improvement in the quality, organization, and completeness of the paper. The authors would also like to thank Sewoong Oh and Kyunghyun Cho for insightful discussions that helped improve the paper.

Appendix A. Proofs of the theorems

In this section, we describe the proofs of all our theoretical results, starting with the results stated in section 4.1.1.

A.1 Analysis of Expected Penalties under the soft-penalty algorithm

First, we introduce some additional notation. Let $\mathbb{P}_{ih}[E]$ (resp. $\mathbb{P}_{ia}[E]$) denote the probability of some event E conditioned on the fact that worker w_i is honest (resp. adversarial). Similarly, $\mathbb{E}_{ih}[\boldsymbol{X}]$ (resp. $\mathbb{E}_{ia}[\boldsymbol{X}]$) denotes the conditional expectation of the random variable $\boldsymbol{X}$ given the event that worker w_i is honest (resp. adversarial). Also, let $f(\cdot)$ denote the probability density function (PDF) of the worker reliability distribution. Recall the definitions $P := q\mu + (1-q)$, $Q := 1 - q\mu$, and the function $g(x) := \frac{1-x^r}{r(1-x)}$, and note that $g(\cdot)$ is *strictly increasing* on $(0,1)$. Let $\mathbf{1}[A]$ denote the indicator variable taking value 1 if A is true and 0 otherwise. Let $\boldsymbol{D}_j^+$ (resp. $\boldsymbol{D}_j^-$) denote the number of workers who label task t_j as $+1$ (resp -1). In other words, $\boldsymbol{D}_j^+ = \sum_{w_i \in W_j} \mathbf{1}[\boldsymbol{W}_{ij} = +1]$ and $\boldsymbol{D}_j^- = \sum_{w_i \in W_j} \mathbf{1}[\boldsymbol{W}_{ij} = -1]$, where recall that W_j denotes the set of workers who labeled task t_j. Here, $\boldsymbol{W}_{ij}$ represents the label assigned by worker w_i to task t_j. Finally, let $\mathrm{Bin}(n,p)$ denote the Binomial distribution with parameters n and p.

We begin by proving some important lemmas that will be repeatedly used in the proofs below.

Lemma 18 *Under the generative model in section 4.1, the probability that worker w_i provides response $+1$ for a task t_j is given by*

$$\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] = P$$
$$\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = -1] = Q$$

Furthermore, conditioned on $\boldsymbol{Y}_j = +1$ or $\boldsymbol{Y}_j = -1$, the random variables $\mathbf{1}[\boldsymbol{W}_{ij} = +1]$ are i.i.d. for all $w_i \in W_j$. As a result, it follows that $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = +1 \sim \mathrm{Bin}(r, P)$ and $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = -1 \sim \mathrm{Bin}(r, Q)$.

Proof Consider the case when $\boldsymbol{Y}_j = +1$:

$$\begin{aligned}
&\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] \\
&= \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] \cdot \Pr[w_i \text{ is honest} \,\big|\, \boldsymbol{Y}_j = +1] \\
&+ \mathbb{P}_{ia}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] \cdot \Pr[w_i \text{ is adversarial} \,\big|\, \boldsymbol{Y}_j = +1] \\
&= \left(\int_0^1 \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = +1] \cdot f(\mu_i) d\mu_i \right) q + 1 \cdot (1-q) \\
&= \left(\int_0^1 \mu_i f(\mu_i) d\mu_i \right) q + (1-q) \\
&= q\mu + (1-q) = P,
\end{aligned}$$

where the first term of the second equality follows from the law of total expectation, the second term because the adversary always labels a task +1, and the third equality follows from the definition of honest worker reliability $\boldsymbol{M}_i$.

Furthermore, it follows from our generative process that conditioned on $\boldsymbol{Y}_j = +1$, the labels from any two workers $w_i \neq w_{i'}$ for task t_j are generated independently. Therefore, we have shown that, conditioned on $\boldsymbol{Y}_j = +1$, the random variables $\mathbf{1}[\boldsymbol{W}_{ij} = +1]$ are independently and identically distributed with probability P of taking the value 1. Because $\boldsymbol{D}_j^+ = \sum_{w_i \in W_j} \mathbf{1}[\boldsymbol{W}_{ij} = +1]$ and $|W_j| = r$, it follows that $\boldsymbol{D}_j^+ \,\Big|\, \boldsymbol{Y}_j = +1$ is a sum of r i.i.d. Bernoulli random variables and, hence, $\boldsymbol{D}_j^+ \,\Big|\, \boldsymbol{Y}_j = +1 \sim \text{Bin}(r, P)$.

A similar argument shows that $\Pr[\boldsymbol{W}_{ij} = +1 \,\Big|\, \boldsymbol{Y}_j = -1] = Q$ and consequently $\boldsymbol{D}_j^+ \,\Big|\, \boldsymbol{Y}_j = -1 \sim \text{Bin}(r, Q)$. ■

Lemma 19 *Under the generative model in section 4.1, suppose that worker w_i is honest with a sampled reliability μ_i and let $\boldsymbol{S}_{ij}$ denote the penalty received by w_i from task $t_j \in \mathcal{T}_i$ in the* SOFT-PENALTY *algorithm. Then, we can show*

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i] = \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i) \cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i) \cdot g(1-Q)\Big)$$

Similarly, if worker w_i is adversarial, then we have:

$$\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] = \gamma \cdot g(1-P) + (1-\gamma) \cdot g(1-Q)$$

Proof We begin with the case when w_i is honest. Using the law of total expectation, we have:

$$\begin{aligned}
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i] \\
&= \sum_{v_1, v_2 \in \{-1,+1\}} \mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = v_1; \boldsymbol{Y}_j = v_2] \cdot \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = v_1; \boldsymbol{Y}_j = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i]
\end{aligned}$$

We first consider the case when $v_1 = +1$ and $v_2 = +1$. In this case, because worker w_i assigns label +1 to task t_j (i.e., $\boldsymbol{W}_{ij} = +1$), the penalty $\boldsymbol{S}_{ij}$ assigned by the task to the worker is equal to $1/\boldsymbol{D}_j^+$. Furthermore, because $\boldsymbol{Y}_j = +1$ and $\mathbf{1}[\boldsymbol{W}_{ij} = +1] = 1$, it follows from the arguments in Lemma 18 that $\boldsymbol{D}_j^+ - 1$ is distributed as $\text{Bin}(r-1, P)$. We can now

write

$$\begin{aligned}
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] \\
=&\mathbb{E}_{ih}\left[1/\boldsymbol{D}_j^+ \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1\right] \\
=&\sum_{k=0}^{r-1} \frac{1}{1+k}\binom{r-1}{k} \cdot P^k(1-P)^{r-1-k} \\
=&\frac{1}{r} \cdot \sum_{k=0}^{r-1} \binom{r}{k+1} \cdot P^k(1-P)^{r-1-k} \\
=&\frac{1}{rP} \cdot \sum_{k'=1}^{r} \binom{r}{k'} \cdot P^{k'}(1-P)^{r-k'} \\
=&\frac{1-(1-P)^r}{rP} = g(1-P),
\end{aligned}$$

where the third equality follows because $\frac{1}{k+1} \cdot \binom{r-1}{k} = \frac{1}{r} \cdot \binom{r}{k+1}$, the fifth equality follows because $\sum_{k'=0}^{r} \binom{r}{k'} \cdot P^{k'}(1-P)^{r-k'} = 1$, and the last equality follows from the definition of the function $g(\cdot)$.

Furthermore, $\mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1, \boldsymbol{Y}_j = +1 \,\Big|\, \boldsymbol{M}_i = \mu_i] = \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1 \,\Big|\, \boldsymbol{Y}_j = +1; \boldsymbol{M}_i = \mu_i] \cdot \mathbb{P}_{ih}[\boldsymbol{Y}_j = +1 \,\Big|\, \boldsymbol{M}_i = \mu_i] = \mu_i \cdot \gamma$. We have thus shown that

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] \cdot \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1 \,\Big|\, \boldsymbol{M}_i = \mu_i] = g(1-P) \cdot \mu_i\gamma$$

The case when $v_1 = -1$ and $v_2 = +1$ (i.e., $\boldsymbol{W}_{ij} = -1$ and $\boldsymbol{Y}_j = +1$) follows a symmetric argument. In particular, because worker w_i assigns the label -1 to task t_j, the penalty $\boldsymbol{S}_{ij}$ that is assigned is equal to $1/\boldsymbol{D}_j^-$. Furthermore, it follows from the arguments in Lemma 18 that $\boldsymbol{D}_j^- - 1$ is distributed as $\mathrm{Bin}(r-1, 1-P)$. It now follows from a symmetric argument (by replacing P by $1-P$ above) that $\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = +1] = g(P)$ and $\mathbb{P}_{ih}[\boldsymbol{Y}_j = +1; \boldsymbol{W}_{ij} = +1 \,\Big|\, \boldsymbol{M}_i = \mu_i] = \gamma \cdot (1-\mu_i)$. We have thus shown that

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = +1] \cdot \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = +1 \,\Big|\, \boldsymbol{M}_i = \mu_i] = g(P) \cdot (1-\mu_i)\gamma$$

Replacing P by Q, μ_i by $1-\mu_i$, and γ by $1-\gamma$ yields the expressions for the other two cases. In particular,

$$\begin{aligned}
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = -1] \cdot \mathbb{P}_{ih}[\boldsymbol{Y}_j = -1; \boldsymbol{W}_{ij} = +1 \,\Big|\, \boldsymbol{M}_i = \mu_i] \\
&= g(1-Q) \cdot (1-\gamma) \cdot (1-\mu_i) \\
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = -1] \cdot \mathbb{P}_{ih}[\boldsymbol{Y}_j = -1; \boldsymbol{W}_{ij} = -1 \,\Big|\, \boldsymbol{M}_i = \mu_i] \\
&= g(Q) \cdot (1-\gamma) \cdot \mu_i
\end{aligned}$$

Combining the above, we obtain

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i] = \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i)\cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i)\cdot g(1-Q)\Big)$$

We now consider the case when worker w_i is an adversary. Because adversaries always assign the label $+1$, we need to consider only two cases: $\boldsymbol{Y}_j = +1$ and $\boldsymbol{Y}_j = -1$. The conditional expectations of the penalties in both cases are identical to those above: $\mathbb{E}_{ia}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] = g(1-P)$ and $\mathbb{E}_{ia}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = -1] = g(1-Q)$. Further, we have $\mathbb{P}_{ia}[\boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] = \gamma$ and $\mathbb{P}_{ia}[\boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = -1] = 1-\gamma$. Combining these, we obtain

$$\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] = \gamma \cdot g(1-P) + (1-\gamma)\cdot g(1-Q)$$

The result of the lemma now follows.

■

We are now ready to prove the theorems.

A.1.1 Proof of Theorem 3

First, note that if $q = 1$ then $P = \mu$ and $Q = 1-\mu$. Also, since all workers are honest, we remove the explicit conditioning on worker w_i being honest. Then the expected penalty allocated to a worker w_i with a reliability μ_i:

$$\begin{aligned}
&\mathbb{E}[\mathbf{PEN}_i \,\Big|\, \boldsymbol{M}_i = \mu_i] = \frac{1}{l}\sum_{j\in\mathcal{T}_i} \mathbb{E}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i] \\
&= \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i)\cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i)\cdot g(1-Q)\Big) \\
&\text{(using lemma 19)} \\
&= \gamma\Big(\mu_i \cdot g(1-\mu) + (1-\mu_i)\cdot g(\mu)\Big) + (1-\gamma)\Big(\mu_i \cdot g(1-\mu) + (1-\mu_i)\cdot g(\mu)\Big) \\
&= g(\mu) - \mu_i \cdot (g(\mu) - g(1-\mu))
\end{aligned}$$

Since $g(\cdot)$ is strictly increasing on $(0,1)$ and $\mu > \frac{1}{2}$, we have that $\mu > 1-\mu$ and consequently $g(\mu) - g(1-\mu) > 0$. The claim then follows.

A.1.2 Proof of Theorem 4

For an honest worker w_i, we have that

$$
\begin{aligned}
&p_h := \mathbb{E}_{ih}[\mathbf{PEN}_i] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\mathbb{E}_{ih}[\boldsymbol{S}_{ij}] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\int_0^1 \mathbb{E}_{ih}[\boldsymbol{S}_{ij}\,\Big|\,\boldsymbol{M}_i=\mu_i]f(\mu_i)d\mu_i \quad \text{(law of total expectation)} \\
&= \int_0^1\Big(\gamma\cdot(\mu_i\cdot g(1-P)+(1-\mu_i)\cdot g(P))+(1-\gamma)\cdot(\mu_i\cdot g(Q)+(1-\mu_i)\cdot g(1-Q))\Big)f(\mu_i)d\mu_i \\
&\text{(using lemma 19)} \\
&= \gamma\Big(\mu\cdot g(1-P)+(1-\mu)\cdot g(P)\Big)+(1-\gamma)\Big(\mu\cdot g(Q)+(1-\mu)\cdot g(1-Q)\Big) \\
&\text{(since } \int_0^1 \mu_i f(\mu_i)d\mu_i=\mu\text{)}
\end{aligned}
$$

Similarly, when worker w_i is adversarial,

$$
\begin{aligned}
p_a &:= \mathbb{E}_{ia}[\mathbf{PEN}_i] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\gamma\cdot g(1-P)+(1-\gamma)\cdot g(1-Q) \quad \text{(using lemma 19)} \\
&= \gamma\cdot g(1-P)+(1-\gamma)\cdot g(1-Q)
\end{aligned}
$$

Suppose $q\mu \le 1/2$, then we have that $Q = 1-q\mu \ge 1/2$ and $P > Q \ge 1/2$. Further, because $g(\cdot)$ is strictly increasing, it follows that $g(1-P)-g(P) < 0$ and $g(1-Q)-g(Q) \le 0$. Given this, and using the expressions for the expected penalty computed above, we have that:

$$
\begin{aligned}
&p_a - p_h \\
&= \gamma\cdot(1-\mu)\cdot\underbrace{(g(1-P)-g(P))}_{<0}+(1-\gamma)\cdot\mu\cdot\underbrace{(g(1-Q)-g(Q))}_{\le 0} \\
&\le 0
\end{aligned}
$$

Therefore, $q\mu > \frac{1}{2}$ is a *necessary* condition for $p_h < p_a$. Assuming this condition is met, we derive the second condition:

$$\begin{aligned}
p_a > p_h &\iff p_a - p_h > 0 \\
&\iff \gamma \cdot (1-\mu) \cdot \Big(g(1-P) - g(P) \Big) + (1-\gamma) \cdot \mu \Big(g(1-Q) - g(Q) \Big) > 0 \\
&\iff (1-\gamma) \cdot \mu \Big(g(1-Q) - g(Q) \Big) > \gamma \cdot (1-\mu) \cdot \Big(g(P) - g(1-P) \Big) \\
&\iff \frac{\mu}{1-\mu} \frac{\Big(g(1-Q) - g(Q) \Big)}{\Big(g(P) - g(1-P) \Big)} > \frac{\gamma}{1-\gamma} \\
&\Big(\text{ since } P > q\mu > \frac{1}{2} \text{ and } Q = 1 - q\mu < \frac{1}{2} \Big)
\end{aligned}$$

Consider the function $h(\mu, q) = \frac{g(1-Q)-g(Q)}{g(P)-g(1-P)}$ in the regime $q\mu > \frac{1}{2}$. Note that as q increases, Q decreases (since $\frac{\partial Q}{\partial q} = -\mu < 0$) and therefore $g(1-Q)-g(Q)$ (strictly) increases. Similarly, P decreases as q increases (since $\frac{\partial P}{\partial q} = \mu - 1 \leq 0$) and therefore $g(P) - g(1-P)$ decreases. It follows that $h(\mu, q)$ is a *strictly* increasing function of q. The result of the theorem now follows.

A.1.3 Proof of Corollary 5

Since $q\mu > \frac{1}{2}$, we have that $P > \frac{1}{2}$ and $Q < \frac{1}{2}$. From the proof of theorem 4 above, we have

$$\begin{aligned}
p_a > p_h &\iff \frac{\mu}{1-\mu} h(\mu, q) > \frac{\gamma}{1-\gamma} \\
&\iff h_\mu(q) > \frac{\gamma}{1-\gamma} \\
&\iff q > h_\mu^{-1} \Big(\frac{\gamma}{1-\gamma} \Big) \\
&\text{(since } h_\mu(q) \text{ is strictly increasing)}
\end{aligned}$$

A.2 Asymptotic identification of honest and adversarial workers

We begin with the proof of the lemma establishing the locally-tree like property of the worker-task assignment graph.

A.2.1 Proof of Lemma 8

We adapt the proof from Karger et al. Consider the following (discrete time) random process that generates the random graph $\mathcal{B}_{\boldsymbol{w},2}$ starting from the root node $\boldsymbol{w}$. In the first step, we connect l task nodes to node $\boldsymbol{w}$ according to the configuration model, where l half-edges are matched to a randomly chosen subset of mr task half-edges of size l. Let α_1 denote the probability that the resulting graph is *not* a tree, that is, at least one pair

of edges are connected to the same task node. Since there are $\binom{l}{2}$ pairs and each pair of half-edges is connected to the same task node with probability $\frac{r-1}{mr-1}$, we have that:

$$\alpha_1 \leq \binom{l}{2}\frac{r-1}{mr-1} \leq \frac{l^2}{2m} = \frac{lr}{2n}$$

where we use the fact that $(a-1)/(b-1) \leq a/b$ for all $a \leq b$ and the relation $mr = nl$. Next, define $\beta_2 \equiv \Pr[\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree} \,\big|\, \mathcal{B}_{\boldsymbol{w},1} \text{ is a tree}]$ so that we have:

$$\Pr\left(\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree}\right) \leq \alpha_1 + \beta_2$$

We can similarly bound β_2. For generating $\mathcal{B}_{\boldsymbol{w},2}$ conditioned on $\mathcal{B}_{\boldsymbol{w},1}$ being a tree, there are $l\hat{r}$ half-edges where $\hat{r} = r-1$. Among the $\binom{l\hat{r}}{2}$ pairs of these half-edges, each pair will be connected to the same worker with probability $\frac{l-1}{l(n-1)-1}$ and therefore:

$$\begin{aligned}\beta_2 &\leq \frac{l^2\hat{r}^2}{2}\frac{l-1}{l(n-1)-1} \\ &\leq \frac{l^2\hat{r}^2}{2}\frac{l}{l(n-1)} = \frac{l^2\hat{r}^2}{2(n-1)}\end{aligned}$$

Combining this with the above, we get that

$$\Pr[\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree}] \leq \alpha_1 + \beta_2 \leq \frac{lr}{2n} + \frac{l^2\hat{r}^2}{2(n-1)} \leq \frac{l^2r^2}{n-1}$$

A.2.2 Proof of Lemma 9

Consider an honest worker w_i with a given reliability μ_i. Recall that the penalty assigned to w_i is of the form:

$$\mathbf{PEN}_i = \frac{1}{l}\sum_{j \in \mathcal{T}_i} \boldsymbol{S}_{ij}$$

where $\boldsymbol{S}_{ij} = \frac{1}{\boldsymbol{D}_j^+}$ if $\boldsymbol{W}_{ij} = +1$ and $\boldsymbol{S}_{ij} = \frac{1}{\boldsymbol{D}_j^-}$ when $\boldsymbol{W}_{ij} = -1$. For any two tasks $t_j \neq t_{j'} \in \mathcal{T}_i$ note that $\boldsymbol{W}_{ij}$ and $\boldsymbol{W}_{ij'}$ are independent, conditioned on the reliability $\boldsymbol{M}_i$. This is because

for $(v_1, v_2) \in \{-1, +1\}$:

$$
\begin{aligned}
&\Pr[\boldsymbol{W}_{ij} = v_1; \boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i] \\
&= \sum_{(x_1,x_2)\in\{-1,+1\}} \Pr[\boldsymbol{W}_{ij} = v_1; \boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2] \Pr[\boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2] \\
&(\text{ since the true task labels are independent of } \boldsymbol{M}_i) \\
&= \Big(\Pr[\boldsymbol{W}_{ij} = v_1 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = +1] \Pr[\boldsymbol{Y}_j = +1] \\
&+ \Pr[\boldsymbol{W}_{ij} = v_1 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = -1] \Pr[\boldsymbol{Y}_j = -1] \Big) \times \\
&\Big(\Pr[\boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = +1] \Pr[\boldsymbol{Y}_{j'} = +1] \\
&+ \Pr[\boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = -1] \Pr[\boldsymbol{Y}_{j'} = -1] \Big) \\
&= \Pr[\boldsymbol{W}_{ij} = v_1 \,|\, \boldsymbol{M}_i = \mu_i] \cdot \Pr[\boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i]
\end{aligned}
$$

Note that third equality makes use of the fact that $\boldsymbol{W}_{ij}$ (resp. $\boldsymbol{W}_{ij'}$) is independent of all other random variables conditioned on the reliability $\boldsymbol{M}_i$ and the true label $\boldsymbol{Y}_j$ (resp. $\boldsymbol{Y}_{j'}$). The argument above can be extended for any subset of random variables $\boldsymbol{W}_{ij_1}, \boldsymbol{W}_{ij_2}, \ldots, \boldsymbol{W}_{ij_i}$. Further, if the worker-task assignment graph is 2-locally tree-like at w_i, there is no other overlap in the set of workers labeling the tasks $t_j, t_{j'}$ apart from w_i. This combined with the above claim shows that the random variables $\{\boldsymbol{S}_{ij} \,:\, j \in \mathcal{T}_i\}$ are *mutually independent* under our generative model. Further, note that $\frac{1}{r} \leq \boldsymbol{S}_{ij} \leq 1$ for any task $t_j \in \mathcal{T}_i$, i.e the random variables $\boldsymbol{S}_{ij}$ are bounded. Then, we can apply Hoeffding's inequality to bound the difference between $\mathbf{PEN}_i$ and $\mathbb{E}[\mathbf{PEN}_i \,\big|\, \boldsymbol{M}_i = \mu_i]$ for any $\varepsilon > 0$:

$$
\Pr\left(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \varepsilon \,\Big|\, \boldsymbol{M}_i = \mu_i \right) \leq \exp\left(\frac{-2l\varepsilon^2}{(1 - 1/r)^2} \right)
$$

A.2.3 Proof of Theorem 6

Suppose we draw a random worker $\boldsymbol{w}$ uniformly from the set of workers W. Then we want to compute the average probability of error, which is the probability that we misclassify a randomly chosen worker:

$$
\frac{1}{n} \sum_{i=1}^{n} \Pr\left(\boldsymbol{I}(w_i) \neq \hat{\boldsymbol{I}}_\theta(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta) \right) = \Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta) \right)
$$

Let $\mathbf{PEN}_{\boldsymbol{w}}$ denote the penalty received by the worker $\boldsymbol{w}$. Recall the expression for the expected penalty received by an honest worker from the proof of theorem 3 above, $\mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \,\big|\, \boldsymbol{M}_{\boldsymbol{w}} = \mu_{\boldsymbol{w}}] = g(\mu) - \mu_{\boldsymbol{w}} \cdot (g(\mu) - g(1-\mu))$. Based on the definition of $\hat{\mu}(\theta)$ in the theorem it follows that $p_{\hat{\mu}(\theta)} := \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \,\big|\, \boldsymbol{M}_{\boldsymbol{w}} = \hat{\mu}(\theta)] = \theta - \varepsilon$.

We upper bound the probability $\Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)\right)$ in two steps. First, if $\mathcal{B}_{\boldsymbol{w},2}$ is not a tree, we suppose that we always declare w_i as adversarial, thereby making an error. So supposing $\mathcal{B}_{\boldsymbol{w},2}$ is a tree, the probability that we misclassify $\boldsymbol{w}$ is given by

$$
\begin{aligned}
&\Pr(\mathbf{PEN}_{\boldsymbol{w}} > \theta \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)) \\
&= \Pr(\mathbf{PEN}_{\boldsymbol{w}} > p_{\hat{\mu}(\theta)} + \varepsilon \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)) \\
&= \int_{\hat{\mu}(\theta)}^{1} \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} > p_{\hat{\mu}(\theta)} + \varepsilon \,\middle|\, \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&\leq \int_{\hat{\mu}(\theta)}^{1} \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} > \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}] + \varepsilon \,\middle|\, \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&(\text{since } \tilde{\mu} > \hat{\mu}(\theta) \implies \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}] < p_{\hat{\mu}(\theta)}) \\
&\leq \int_{\hat{\mu}(\theta)}^{1} \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&(\text{from lemma 9 above}) \\
&\leq \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
\end{aligned}
$$

Finally, combining all the claims above, we have

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\boldsymbol{I}(w_i) \neq \hat{\boldsymbol{I}}_\theta(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) \\
&= \Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)\right) \\
&\leq \Pr(\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree }) \cdot 1 + \Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta) \,\middle|\, \mathcal{B}_{\boldsymbol{w},2} \text{ is a tree }\right) \cdot 1 \\
&\leq \frac{l^2 r^2}{n-1} + \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
\end{aligned}
$$

and the first part of the theorem follows. For the second part, when $l = \log n$ and r is fixed, note that

$$
\exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) = \exp\left(\frac{-2\log n\varepsilon^2}{(1-1/r)^2}\right) = \exp\left(\log\left(\frac{1}{n}\right)^{\frac{2\varepsilon^2}{(1-1/r)^2}}\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right)
$$

In addition, because $\varepsilon < \frac{1}{\sqrt{2}} \implies 2\varepsilon^2 < 1$, it follows that $\frac{\log^2 n \cdot r^2}{n-1} = O\left(\frac{1}{n^{2\varepsilon^2}}\right)$ and the claim follows.

A.2.4 Proof of Theorem 10

We follow a similar line of reasoning to that of theorem 6. As before, if $\mathcal{B}_{\boldsymbol{w},2}$ is not a tree, we suppose that the worker is always misclassified.

First, we focus on **honest** workers. Given a threshold θ, choose the reliability threshold such that $\mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \hat{\mu}(q,\theta)] = \theta - \varepsilon$ where the expected penalty for an honest worker

conditioned on reliability was computed in lemma 19. Specifically, we have the following expression for $\hat{\mu}(q,\theta)$:

$$\begin{aligned}
&\gamma\Big(\hat{\mu}(q,\theta)\cdot g(1-P)+(1-\hat{\mu}(q,\theta))\cdot g(P)\Big)+(1-\gamma)\Big(\hat{\mu}(q,\theta)\cdot g(Q)+(1-\hat{\mu}(q,\theta))\cdot g(1-Q)\Big)\\
&=\theta-\varepsilon \implies\\
&\Big(\gamma g(1-P)-\gamma g(P)+(1-\gamma)g(Q)-(1-\gamma)g(1-Q)\Big)\hat{\mu}(q,\theta)\\
&=\theta-\varepsilon-\gamma g(P)-(1-\gamma)g(1-Q) \implies\\
&\hat{\mu}(q,\theta)=\frac{\gamma g(P)+(1-\gamma)g(1-Q)+\varepsilon-\theta}{\gamma\cdot(g(P)-g(1-P))+(1-\gamma)\cdot(g(1-Q)-g(Q))}
\end{aligned}$$

Note that such a threshold always exists since (1) $\theta-\varepsilon > p_h = \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i=\mu]$ (2) $\mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i=\mu_i]$ (strictly) increases as μ_i decreases and (3) $\mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i=0] \geq p_a > \theta > \theta-\varepsilon$. In particular, this means that $\hat{\mu}(q,\theta) < \mu$. Further, observe that if $q=1$, then we have $\hat{\mu}(1,\theta)=\hat{\mu}(\theta)$.

Then, the probability that we misclassify a randomly chosen honest worker is given by

$$\Pr(\mathbf{PEN}_{\boldsymbol{w}} > \theta \,\Big|\, \boldsymbol{w} \text{ is honest }) \leq \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} \geq \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}}=\hat{\mu}(q,\theta)]+\varepsilon \,\Big|\, \boldsymbol{w} \text{ is honest }\right)$$

The claim then follows from the result of lemma 11 below.

When $\boldsymbol{w}$ is **adversarial**, the probability that we misclassify $\boldsymbol{w}$ is given by

$$\Pr(\mathbf{PEN}_{\boldsymbol{w}} \leq \theta \mid \boldsymbol{w} \text{ is adversarial}) \leq \Pr(\mathbf{PEN}_{\boldsymbol{w}} \leq p_a-\varepsilon \mid \boldsymbol{w} \text{ is adversarial}) \leq \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)$$

where the first inequality follows since $\theta < p_a-\varepsilon$ and the second follows from the result of lemma 11 below.

Coming to the second part of the theorem, first observe that the expected penalties p_h and p_a lie between 0 and 1. As a result, we have that $\varepsilon < (p_a-p_h)/2 \leq \frac{1}{2} \implies 2\varepsilon^2 < 1$ and the claim follows from the sequence of arguments in the proof of theorem 6 above.

A.2.5 Proof of Lemma 11

Suppose that worker w_i is **honest** and let $p_{\hat{\mu}} = \mathbb{E}[\mathbf{PEN}_i \,\big|\, \boldsymbol{M}_i = \hat{\mu}]$. We have the following (conditioned on $\mathcal{B}_{w_i,2}$ being a tree):

$$
\begin{aligned}
&\Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest }) \\
&= \int_0^1 \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&= \int_0^{\hat{\mu}} \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&+ \int_{\hat{\mu}}^1 \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&\leq \int_0^{\hat{\mu}} f(\mu_i) d\mu_i \\
&+ \int_{\hat{\mu}}^1 \Pr(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&(\text{ since } \mu_i \geq \hat{\mu} \implies \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] \leq p_{\hat{\mu}}) \\
&\leq F(\hat{\mu}) + \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) \\
&(\text{ from lemma 9})
\end{aligned}
$$

When w_i is an **adversary**, note that her responses $\{\boldsymbol{W}_{ij} \,:\, j \in \mathcal{T}_i\}$ are trivially independent since adversarial workers always respond with $+1$ on all assigned tasks. Further, since $\mathcal{B}_{w_i,2}$ is locally tree-like, there is no other overlap in the set of workers labeling any two tasks $t_j, t_{j'} \in \mathcal{T}_i$ apart from w_i. As a result, the assigned penalties $\{\boldsymbol{S}_{ij} \,:\, j \in \mathcal{T}_i\}$ are *mutually independent* and using the Hoeffding's inequality we can establish the concentration bound for adversarial workers:

$$
\Pr[\mathbf{PEN}_i \leq p_a - \varepsilon \,\big|\, w_i \text{ is adversarial}] \leq \exp\left(\frac{-2l^2\varepsilon^2}{\sum_{j=1}^{l}(1-1/r)^2}\right) = \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
$$

A.3 Extension to random, normalized hard-penalty

Here, we show that the theoretical results proved above for the soft-penalty algorithm extend to the random, normalized variant of the hard-penalty algorithm mentioned in section 3.3. First, we focus on the expected penalties. Since the penalty algorithm is randomized, the expectation also takes into account the randomness in the algorithm. As above, if $\boldsymbol{S}_{ij}$ denotes the penalty received by worker w_i from task t_j, then we have that $\boldsymbol{S}_{ij} \in \{0, 1\}$. Further, conditioned on the fact that $\boldsymbol{W}_{ij} = +1$, we have that $\mathbb{E}[\boldsymbol{S}_{ij} \mid \boldsymbol{W}_{ij} = +1] = \mathbb{E}[\frac{1}{\boldsymbol{D}_j^+} \mid \boldsymbol{W}_{ij} = +1]$ using the law of iterated expectations. Similarly, for the case when $\boldsymbol{W}_{ij} = -1$. Then, using the arguments above it is easy to see that the expressions for the expected penalties are the same.

Moving on to the concentration results, observe that $\boldsymbol{S}_{ij}$ depends *only* on $\boldsymbol{W}_{ij}$ and $\boldsymbol{D}_j^+$ (and consequently, $\boldsymbol{D}_j^-$). This is because when $\boldsymbol{W}_{ij} = +1$, we have

$$(\boldsymbol{S}_{ij} \mid \boldsymbol{W}_{ij} = +1) = \begin{cases} 1 & \text{w.p. } 1/\boldsymbol{D}_j^+ \\ 0 & \text{w.p. } 1 - 1/\boldsymbol{D}_j^+ \end{cases}$$

Similarly, for the case when $\boldsymbol{W}_{ij} = -1$. Now, when $\mathcal{B}$ is locally tree-like at worker node w_i, the arguments in lemma 9 and 11 establish that the random variables $\{\boldsymbol{S}_{ij} : t_j \in \mathcal{T}_i\}$ are still mutually independent conditioned on the identity of worker w_i (this also relies on the fact that each task is treated independently when computing the random semi-matching). Therefore, we can apply the Hoeffding's bound to establish the concentration of the penalties under the random, normalized variant of the hard-penalty algorithm around the expected values.

A.4 Proof of Lemma 13

For the simple majority algorithm, it is easy to see that each task has k incorrect responses and at most $r < k$ correct responses and consequently, it outputs the incorrect label for all tasks.

For the expectation-maximization (EM) algorithm, recall that the generative model for worker ratings was as follows: the true task labels are sampled from a population with $\gamma \in [0, 1]$ fraction of tasks having $+1$ true label. Each worker w_i has reliability parameter $\mu_i \in [0, 1]$, which specifies the probability that w_i provides the correct label on any task. Then, given the response matrix $\mathcal{L}$, the log-likelihood of the parameters, $\boldsymbol{\Theta} = (\mu_1, \mu_2, \cdots, \mu_n, \gamma)$ under this generative model can be written as:

$$\log \Pr[\mathcal{L} \mid \boldsymbol{\Theta}] = \sum_{j=1}^{m} \log \left(a_j \cdot \gamma + b_j \cdot (1 - \gamma) \right)$$

where $M(j)$ is the set of workers who label task t_j and

$$a_j = \prod_{i \in M(j)} {\mu_i}^{\mathbf{1}[\mathcal{L}_{ij} = +1]} \cdot (1 - \mu_i)^{\mathbf{1}[\mathcal{L}_{ij} = -1]}$$
$$b_j = \prod_{i \in M(j)} (1 - \mu_i)^{\mathbf{1}[\mathcal{L}_{ij} = +1]} \cdot {\mu_i}^{\mathbf{1}[\mathcal{L}_{ij} = -1]}$$

The objective is to find the model parameters $\boldsymbol{\Theta}^*$ that maximize the log-likelihood, i.e. $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \log \Pr[\mathcal{L} \mid \Theta]$. The EM algorithm computes the maximum likelihood estimates (MLE) of the model parameters by introducing the latent true label of each task, denoted by the vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$. The complete data log-likelihood can then be written as:

$$\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \boldsymbol{\Theta}] = \sum_{j=1}^{m} \Big(y_j \log(a_j \gamma) + (1 - y_j) \log(1 - \gamma) b_j \Big) \tag{1}$$

Each iteration x of the EM algorithm consists of two steps:

- **E-step:** Given the response matrix $\mathcal{L}$ and the current estimate of the model parameters $\mathbf{\Theta}^{(x)}$, the conditional expectation of the complete data log-likelihood is computed as

$$\mathbb{E}\left\{\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \mathbf{\Theta}^{(x)}]\right\} = \sum_{j=1}^{m} \left(\gamma_j^{(x)} \log(a_j^{(x)}\gamma^{(x)}) + (1-\gamma_j^{(x)}) \log(1-\gamma^{(x)}) b_j^{(x)}\right)$$

  where the expectation is w.r.t to $\Pr[\boldsymbol{y} \mid \mathcal{L}; \mathbf{\Theta}^{(x)}]$ and $\gamma_j^{(x)} = \Pr[y_j = +1 \mid \mathcal{L}; \mathbf{\Theta}^{(x)}]$. Using Bayes theorem, we can compute

$$\begin{aligned} \gamma_j^{(x)} &= \frac{\Pr[\{\mathcal{L}_{ij} \;\; : \;\; i \in M(j)\} \mid y_j = +1; \mathbf{\Theta}^{(x)}] \cdot \Pr[y_j = +1 \mid \mathbf{\Theta}^{(x)}]}{\Pr[\{\mathcal{L}_{ij} \;\; : \;\; i \in M(j)\} \mid \mathbf{\Theta}^{(x)}]} \\ &= \frac{a_j^{(x)}\gamma^{(x)}}{a_j^{(x)}\gamma^{(x)} + b_j^{(x)}(1-\gamma^{(x)})} \end{aligned} \tag{2}$$

  where the first equality follows from the fact that labels for different tasks t_j are independent of each other and the second equality follows from the law of total probability.

- **M-step:** Based on the current posterior estimates of the true labels $\gamma_j^{(x)}$ and the response matrix $\mathcal{L}$, the model parameters are updated by maximizing $\mathbb{E}\left\{\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \mathbf{\Theta}^{(x)}]\right\}$, which can be shown to be a lower bound on the data log-likelihood (equation (1)). The prevalence of positive tasks γ is updated as:

$$\gamma^{(x+1)} = \frac{\sum_{j=1}^{m} \gamma_j^{(x)}}{m} \tag{3}$$

  Similarly, the parameter μ_i is updated as:

$$\mu_i^{(x+1)} = \frac{\sum_{j \in N(i)} \left(\mathbf{1}[\mathcal{L}_{ij} = +1]\gamma_j^{(x)} + \mathbf{1}[\mathcal{L}_{ij} = -1](1-\gamma_j^{(x)})\right)}{|N(i)|} \tag{4}$$

  where $N(i)$ is the set of tasks assigned to worker w_i.

These two steps are iterated until convergence of the log-likelihood $\log \Pr[\mathcal{L} \mid \mathbf{\Theta}]$. After convergence, the true labels of the tasks are estimated as:

$$\hat{y}_j = 2 \cdot \mathbf{1}[\gamma_j^{(K)} \geq 0.5] - 1 \tag{5}$$

where K is the number of iterations to convergence.

Since the original problem is non-convex, the EM algorithm converges to a local optimum in general. Its performance critically depends on the initialization of the posterior estimates $\gamma_j^{(0)}$ for each task t_j. A popular approach is to use the majority estimate (see Raykar and Yu (2012)), given by $\gamma_j^{(0)} := \frac{\sum_{i \in M(j)} \mathbf{1}[\mathcal{L}_{ij}=+1]}{|M(j)|}$. We show that this initialization results in incorrect labels for all the tasks.

Suppose that the true labels for all tasks are $+1$, i.e $y_j = +1$ for all $1 \leq j \leq m$. Then all honest worker labels are $+1$ and adversary labels are -1. Further, since each task has

more adversary than honest labels, we have that $\gamma_j^{(0)} < 0.5$ for all $1 \leq j \leq m$. Given this, it easy to see from equation (4) that $\mu_i^{(1)} < 0.5$ whenever w_i is honest and $\mu_i^{(1)} > 0.5$ for all adversarial workers w_i. In addition, we have $\gamma^{(1)} < 0.5$ which follows from equation (3). With these estimates, we update the posterior probabilities γ_j for each task t_j. Again, it follows that $\gamma_j^{(1)} < 0.5$ for all $1 \leq j \leq m$, using the update rule in equation (2). The above sequence of claims shows that $\gamma_j^{(x)} < 0.5$ for all iterations $x \geq 1$ and consequently, $\hat{y}_j = -1$ for all tasks t_j (using equation (5)). Therefore, the EM algorithm outputs incorrect labels for all tasks.

Now, let us consider the general case. If we initialize using the majority estimate, we obtain that $\gamma_j^{(0)} > 0.5$ for all tasks with true label $y_j = -1$ and $\gamma_j^{(0)} < 0.5$ for all tasks with $y_j = +1$. Then, it follows from equation (4) that $\mu_i^{(1)} < 0.5$ for all honest workers w_i and $\mu_i^{(1)} > 0.5$ for adversarial workers w_i. In addition, if we look at the update equations (3) and (4) together, we also get that $\mu_i^{(1)} \geq \max(1-\gamma^{(1)}, \gamma^{(1)})$ for all adversaries w_i. To see this, first note that adversaries label on all the tasks, so that $|N(i)| = m$ for an adversarial worker w_i. Next, let $N_+(i)$ and $N_-(i)$ denote the set of tasks for which adversary w_i labels $+1$ and -1 respectively. If $N_+(i)$ is empty, then we have $\mu_i^{(1)} = 1-\gamma^{(1)}$. Also, since we have all adversary labels as -1, we have $\gamma_j^{(0)} < 0.5$ for all $1 \leq j \leq m$ and therefore $\gamma^{(1)} < 1-\gamma^{(1)}$. Therefore, in this case we have $\mu_i^{(1)} \geq \max(1-\gamma^{(1)}, \gamma^{(1)})$. A symmetric argument can be applied when $N_-(i)$ is empty. So suppose that both $N_+(i)$ and $N_-(i)$ are non-empty, then we have from equation (4):

$$
\begin{aligned}
\mu_i^{(1)} &= \frac{\sum_{j \in N(i)} \left(\mathbf{1}[\mathcal{L}_{ij} = +1]\gamma_j^{(0)} + \mathbf{1}[\mathcal{L}_{ij} = -1](1-\gamma_j^{(0)}) \right)}{|N(i)|} \\
&= \frac{\sum_{j \in N_+(i)} \gamma_j^{(0)} + \sum_{j \in N_-(i)} (1-\gamma_j^{(0)})}{m} \\
&> \frac{\sum_{j \in N_+(i)} \gamma_j^{(0)} + \sum_{j \in N_-(i)} \gamma_j^{(0)}}{m} \\
&\text{(since } \gamma_j^{(0)} < 0.5 \text{ if adversaries label } t_j \text{ as } -1) \\
&= \frac{\sum_{j \in N(i)} \gamma_j^{(0)}}{m} \\
&= \gamma^{(1)}
\end{aligned}
$$

where the last equality follows from equation (3). Similarly, we can show that $\mu_i^{(1)} > 1-\gamma^{(1)}$ and the claim follows.

Now, let $M_h(j)$ and $M_a(j)$ denote the set of honest and adversarial workers labeling task t_j. Consider a task t_j for which the true label $y_j = -1$. Now, we have two cases:

Case 1: $\gamma^{(1)} \geq 0.5$. First observe that since $\mu_i^{(1)} < 0.5$ for all honest workers w_i, we have $\mu_i^{(1)} < 1 - \mu_i^{(1)}$ for all honest workers w_i. Similarly, we have $1 - \mu_i^{(1)} < \mu_i^{(1)}$ for all

adversarial workers w_i. Then, we have

$$\begin{aligned} b_j^{(1)} \cdot (1-\gamma^{(1)}) &= \left(\prod_{i \in M(j)} (1-\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \right) \cdot (1-\gamma^{(1)}) \\ &= \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot \prod_{i \in M_a(j)} (1-\mu_i^{(1)}) \cdot (1-\gamma^{(1)}) \\ &\quad \text{(since honest worker response is } -1 \text{ and adversary response is } +1) \\ &< \prod_{i \in M_h(j)} (1-\mu_i^{(1)}) \cdot \prod_{i \in M_a(j)} \mu_i^{(1)} \cdot \gamma^{(1)} \\ &= a_j^{(1)} \cdot \gamma^{(1)} \end{aligned}$$

where the inequality follows from the observation above and the fact that $\gamma^{(1)} \geq 0.5$. Using equation (2), it follows that $\gamma_j^{(1)} > 0.5$.

Case 2: $\gamma^{(1)} < 0.5$. In this scenario, we have:

$$\begin{aligned} & b_j^{(1)} \cdot (1-\gamma^{(1)}) \\ &= \left(\prod_{i \in M(j)} (1-\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \right) \cdot (1-\gamma^{(1)}) \\ &= \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot \prod_{i \in M_a(j)} (1-\mu_i^{(1)}) \cdot (1-\gamma^{(1)}) \\ &= \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot (1-\mu_{w_{i_1}}^{(1)}) \cdot (1-\gamma^{(1)}) \cdot \prod_{i \in M_a(j) \setminus \{i_1\}} (1-\mu_i^{(1)}) \\ &\quad \text{(where } w_{i_1} \text{ is some adversarial worker)} \\ &\leq \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot \mu_{w_{i_1}}^{(1)} \cdot \gamma^{(1)} \cdot \prod_{i \in M_a(j) \setminus \{i_1\}} (1-\mu_i^{(1)}) \\ &\quad \Big(\text{since } 1-\gamma^{(1)} \leq \mu_{w_{i_1}}^{(1)} \text{ because } w_{i_1} \text{ is adversarial; see discussion before Case 1} \Big) \\ &< \prod_{i \in M_h(j)} (1-\mu_i^{(1)}) \cdot \mu_{w_{i_1}}^{(1)} \cdot \gamma^{(1)} \cdot \prod_{i \in M_a(j) \setminus \{i_1\}} \mu_i^{(1)} \\ &\quad \text{(from Case 1 above)} \\ &= a_j^{(1)} \cdot \gamma^{(1)} \end{aligned}$$

Consequently, in both cases, the posterior probability $\gamma_j^{(1)} > 0.5$ given that the true label was $y_j = -1$. A symmetric argument shows that $\gamma_j^{(1)} < 0.5$ if $y_j = +1$. Then, the above sequence of claims establishes that this remains true for all iterations in the future. Finally, step (5) implies that the output label is incorrect for all tasks.

A.5 Proof of Theorem 14

We prove the result for the the case when there exists at least one subset $\mathcal{T}' \subseteq \mathcal{T}$ such that $\text{PreIm}(\mathcal{T}') \leq k$. Otherwise, the lower bound $L = 0$ by definition and the result of the theorem is trivially true.

Let H^* denote the set $\text{PreIm}(\mathcal{T}^*)$ where

$$\mathcal{T}^* \stackrel{\text{def}}{=} \underset{\mathcal{T}' \subseteq \mathcal{T}:\, |\text{PreIm}(\mathcal{T}')| \leq k}{\arg\max} \left|\mathcal{T}'\right|.$$

We construct an adversary strategy σ^* under which at least L tasks are affected for some true labeling of the tasks, for any decision rule $R \in \mathcal{C}$. Specifically for a fixed honest worker-task assignment graph $\mathcal{B}_H$ and ground-truth labeling $\boldsymbol{y}$ of the tasks, consider the following adversary strategy (that depends on the obtained honest worker responses): letting $H^* = \left\{h_1, h_2, \ldots, h_{|H^*|}\right\}$ and the set of adversaries $A = \{a_1, a_2, \ldots, a_k\}$, we have (recall the notation in Section 2)

$$a_i(t) = \begin{cases} -h_i(t) & \text{if } t \in \mathcal{T}^* \\ h_i(t) & \text{otherwise} \end{cases} \qquad \forall i = 1, 2, \ldots, |H^*| \tag{6}$$

In other words, the adversaries label *flip* the labels of the honest workers H^* for tasks in $\mathcal{T}^*$ and *copy* their responses for all other tasks. Note that since $|H^*| \leq k$ by construction, the above strategy is feasible. In addition, if $|H^*| < k$, then we only use $|H^*|$ of the k adversary identities. Let $\mathcal{L}(\boldsymbol{y})$ denote the $n \times m$ labeling matrix obtained for this adversary strategy, where we explicitly denote the dependence on the true label vector $\boldsymbol{y}$. Here n denotes the total number of workers.

Now consider the scenario in which the true labels of all tasks in $\mathcal{T}^*$ were reversed, let this ground-truth be denoted as $\tilde{\boldsymbol{y}}$. Let $\tilde{h}(t)$ denote the response of honest worker h for task t in the new scenario. Since, honest workers always respond correctly, we have that:

$$\tilde{h}(t) = \begin{cases} -h(t) & \text{if } t \in \mathcal{T}^* \\ h(t) & \text{otherwise} \end{cases} \qquad \forall\ h \in H \tag{7}$$

Correspondingly, according to the adversary labeling strategy σ^* described above, the adversary responses would also change. In particular, using $\tilde{a}_i(t)$ to denote the adversary response in the new scenario, we have

$$\tilde{a}_i(t) = \begin{cases} -\tilde{h}_i(t) & \text{if } t \in \mathcal{T}^* \\ \tilde{h}_i(t) & \text{otherwise} \end{cases} \qquad \forall i = 1, 2, \ldots, |H^*| \tag{8}$$

Finally, let $\mathcal{L}(\tilde{\boldsymbol{y}})$ denote the labeling matrix in this new scenario. We now argue that $\mathcal{L}(\tilde{\boldsymbol{y}}) = P\mathcal{L}(\boldsymbol{y})$ for some $n \times n$ permutation matrix P. In order to see this, for any worker w (honest or adversary), let $r(w)$ and $\tilde{r}(w)$ respectively denote the row vectors in matrices $\mathcal{L}(\boldsymbol{y})$ and $\mathcal{L}(\tilde{\boldsymbol{y}})$. We show that $\mathcal{L}(\tilde{\boldsymbol{y}})$ can be obtained from $\mathcal{L}(\boldsymbol{y})$ through a permutation of the rows.

First observe that for any honest worker $h \notin H^*$, we must have by definition of PreIm that $h(t) = 0$ for any $t \in \mathcal{T}^*$. Thus, it follows from equation (7) that $\tilde{h}(t) = -h(t) = 0 = h(t)$

for any $t \in \mathcal{T}^*$. Furthermore, $\tilde{h}(t) = h(t)$ for any $t \notin \mathcal{T}^*$ by (7). Therefore, we have that $\tilde{r}(h) = r(h)$ for any $h \notin H^*$. Next, consider an honest worker $h_i \in H^*$ for some i. It can be argued that $\tilde{r}(h_i) = r(a_i)$. To see this, for any task $t \notin \mathcal{T}^*$, we have by equation (7) that $\tilde{h_i}(t) = h_i(t) = a_i(t)$, where the second equality follows from equation (6). Similarly, for any $t \in \mathcal{T}^*$, we have $\tilde{h_i}(t) = -h_i(t) = a_i(t)$ again using equation (6). Thus, we have shown that the rows $\tilde{r}(h_i) = r(a_i)$ for any $1 \leq i \leq |H^*|$. A symmetric argument shows that $\tilde{r}(a_i) = r(h_i)$ for all $1 \leq i \leq |H^*|$. Consequently, $\mathcal{L}(\tilde{\boldsymbol{y}})$ is obtained from $\mathcal{L}(\boldsymbol{y})$ by swapping rows corresponding to h_i with a_i for all $i = 1, 2, \ldots, |H^*|$.

Now that we have shown $\mathcal{L}(\tilde{\boldsymbol{y}}) = P\mathcal{L}(\boldsymbol{y})$ for some permutation matrix P, it follows from the fact that $R \in \mathcal{C}$ that $R(\mathcal{L}(\tilde{\boldsymbol{y}})) = R(\mathcal{L}(\boldsymbol{y}))$. Thus, the labels output by R for all tasks in $\mathcal{T}^*$ is the same under both scenarios. As a result, it follows that $\text{Aff}(R, \sigma^*, \boldsymbol{y}) + \text{Aff}(R, \sigma^*, \tilde{\boldsymbol{y}}) = |\mathcal{T}^*| = 2 * L$ and therefore, either $\text{Aff}(R, \sigma^*, \boldsymbol{y}) \geq L$ or $\text{Aff}(R, \sigma^*, \tilde{\boldsymbol{y}}) \geq L$.

In other words, there exists a ground-truth task labeling for which the number of affected tasks is at least L, and since we take a maximum over all possible ground-truth labelings, the result of the theorem follows. Further, any decision rule that outputs labels randomly in case of ties (i.e equal number of $+1$ and -1 responses) will achieve the lower bound L, including the simple majority decision rule.

Misclassification of workers. Since we allow the sophisticated adversaries to adopt arbitrary strategies, we need to assume some characteristic property that helps to identify honest workers. In the theorem we assumed that all honest workers are perfectly reliable, so that they agree with each other on their labels for all tasks. Given this, any ML algorithm that is trying to separate honest workers from the sophisticated adversaries will output two groups of workers, say W_1, W_2 such that $W_1 \cup W_2 = H \cup A$ and $W_1 \cap W_2 = \phi$. Further, let us suppose that the algorithm has knowledge of k—the number of sophisticated adversaries, so that the output satisfies $|W_1| = n - k$ and $|W_2| = k$ where $|H \cup A| = n$. We call an output (W_1, W_2) *valid* if all workers in W_1 agree with each other on their labels for all tasks. Note that the "true" output $W_1 = H$ and $W_2 = A$ is valid. We argue that for the above adversary strategy, the output $W_1 = (H \setminus H^*) \cup A$ and $W_2 = H^*$ is also valid, where H^* is as defined in the proof above.

To see this, note that the algorithm can identify the set $\mathcal{T}^*$ of tasks for which there are conflicts, i.e. both $+1$ and -1 labels, and partition the set of workers labeling these tasks into two groups where workers in each group agree on their label for all tasks. Now, one of these groups consists of adversarial workers A and the other honest workers, corresponding to the set H^* above. The remaining workers correspond to the set $H \setminus H^*$. Now, if we consider the output $W_1 = (H \setminus H^*) \cup A$, any two workers in W_1 indeed agree with each other on their labels for all tasks. This is because, by the definition of PreIm, workers in $H \setminus H^*$ *only* label tasks in the set $\mathcal{T} \setminus \mathcal{T}^*$. Further, since adversaries agree with honest workers H^* on these tasks (refer to the strategy above) and honest workers are perfectly reliable, this means that adversaries A and honest workers $H \setminus H^*$ also agree with each other on their labels for all tasks in $\mathcal{T} \setminus \mathcal{T}^*$. Consequently, we have that the output $((H \setminus H^*) \cup A, H^*)$ is also *valid*. Further note that this is the only other valid output, since honest workers H^* and adversaries A do *not* agree with each other on their labels for tasks in $\mathcal{T}^*$ and therefore, cannot be placed together in the set W_1.

Finally, since the ML algorithm does not have any other information, it cannot distinguish between the two valid outputs described above, so that a random guess will misclassify $2k$ workers with probability $(1/2)$.

A.5.1 Proof of Corollary 16

We first prove that $2L \geq \lfloor \frac{k}{r} \rfloor$. Consider $\mathcal{T}' \subseteq \mathcal{T}$ such that $|\mathcal{T}'| = \lfloor \frac{k}{r} \rfloor$, note that we can always choose such a $\mathcal{T}'$ since $k < |H| \leq r \cdot |\mathcal{T}| \implies k/r < |\mathcal{T}|$. Since $\mathcal{B}_H$ is r-right regular, the pre-image of $\mathcal{T}'$ in $\mathcal{B}_H$ satisfies $|\text{PreIm}(\mathcal{T}')| \leq r|\mathcal{T}'| = r\lfloor \frac{k}{r} \rfloor \leq k$. In other words, any subset of tasks of size $\lfloor \frac{k}{r} \rfloor$ has a pre-image of size at most k. By the definition of L, we have that $2L \geq \lfloor \frac{k}{r} \rfloor$.

For the upper-bound, consider any $\mathcal{T}' \subset \mathcal{T}$ such that $|\mathcal{T}'| = e$ where $e := \lceil \frac{k}{\alpha} \rceil + 1$. Since $e < \gamma|\mathcal{T}|$, by the expander property we have that $|PreIm(\mathcal{T}')| \geq \alpha|\mathcal{T}'| = \alpha \cdot e \geq \alpha \cdot (\frac{k}{\alpha} + 1) > k$. This means that any subset $\mathcal{T}'$ of size *at least* e has a pre-image of size strictly greater than k, since the size of the pre-image can only increase with the addition of more tasks. Therefore, this implies that $2L \leq \lceil \frac{k}{\alpha} \rceil$.

For the second part of the corollary, refer to Theorem 4.4 in Chapter 4 of Vadhan et al. (2012).

A.6 Proof of Theorem 17

Before we can prove the theorem, we need the following definitions and lemmas.

Definition 20 *A bipartite graph $G = (V_1, V_2, E)$ is termed* **degenerate** *if the following condition is satisfied:*

$$|V_1| > |V_2|$$

Definition 21 *A bipartite graph $G = (V_1, V_2, E)$ is termed* **growth** *if the following condition is satisfied:*

$$\forall \ V \subseteq V_1, \ \ |V| \leq |\text{Img}(V)|$$

where $\text{Img}(V) = \{v_2 \in V_2 \mid \exists v \in V \ \text{ s.t. } (\text{v}, \text{v}_2) \in \text{E}\}$*, i.e. the set of neighboring nodes of V.*

Lemma 22 *Any bipartite graph can be decomposed into degenerate and growth sub-graphs where there are cross-edges only between the left nodes of the growth component and the right nodes of the degenerate component.*

Proof Let $G = (V_1, V_2, E)$ be a given bipartite graph. Define V^* to be the largest subset of V_1 such that $|V^*| > |\text{Img}_G(V^*)|$ where Img_G denotes the image in the graph G. If no such V^* exists then the graph is already growth and we are done. If $V^* = V_1$ then the graph is degenerate and again we are done. Else, we claim that the sub-graph J of G restricted to $V_1 \setminus V^*$ on the left and $V_2 \setminus \text{Img}_G(V^*)$ on the right is growth. Suppose not, then there exists a subset V' of nodes on the left such that $|V'| > |\text{Img}_J(V')|$ where $\text{Img}_J(V') \subseteq V_2 \setminus \text{Img}_G(V^*)$ denotes the image of V' in the sub-graph J. But then, we can add V' to V^* to get a larger degenerate sub-graph in G which contradicts our choice of

V^*. To see this, consider the set $V^* \cup V'$ on the left and $\text{Img}_G(V^*) \cup \text{Img}_J(V')$ on the right. We have $|V^* \cup V'| = |V^*| + |V'| > |\text{Img}_G(V^*)| + |\text{Img}_J(V')| = |\text{Img}_G(V^* \cup V')|$. Also, note that the only cross-edges are between $V_1 \setminus V^*$ and $\text{Img}_G(V^*)$. The claim then follows. ■

Lemma 23 *Let $G = (V_1, V_2, E)$ be a bipartite graph and suppose that M is* any *semi-matching on G. Further, let $J = (V_1, V_2', E')$ be the subgraph of G restricted to only the nodes $V_2' \subseteq V_2$ on the right. Starting with $M' \subseteq M$, we can use algorithm $\mathcal{A}_{SM2}$ in Harvey et al. (2003) to obtain an optimal semi-matching $\mathcal{N}$ for the subgraph J. Let the nodes in V_1 be indexed such that $deg_M(1) \geq deg_M(2) \geq \ldots \geq deg_M(|V_1|)$ and indexed again such that $deg_{\mathcal{N}}(1) \geq deg_{\mathcal{N}}(2) \geq \ldots \geq deg_{\mathcal{N}}(|V_1|)$. Then for any $1 \leq s \leq |V_1|$, we have $\sum_{i=1}^{s} deg_{\mathcal{N}}(i) \leq \sum_{i=1}^{s} deg_M(i)$, i.e. the sum of the top s-degrees can only decrease as we go from M to $\mathcal{N}$.*

Proof Note that if we restrict M to just the nodes V_2', we get a feasible semi-matching M' on the subgraph J. Algorithm $\mathcal{A}_{SM2}$ proceeds by the iterated removal of cost-reducing paths. Note that when a cost-reducing path is removed, load is transferred from a node with larger degree (in the current semi-matching) to a node with strictly smaller degree. To see this, let $P = (v_1^{(1)}, v_2^{(1)}, v_1^{(2)}, \ldots, v_1^{(d)})$ be a cost-reducing path (see section 2.1 in Harvey et al. (2003)) in some semi-matching $\bar{M}$ on J. This means that $deg_{\bar{M}}(v_1^{(1)}) > deg_{\bar{M}}(v_1^{(d)}) + 1$. When we eliminate the cost-reducing path P, the degree of $v_1^{(1)}$ decreases by 1 and that of $v_1^{(d)}$ increases by 1, but still the new degree of $v_1^{(d)}$ is strictly lower than the old degree of $v_1^{(1)}$. In other words, if $d_1^{\text{bef}} \geq d_2^{\text{bef}} \geq \ldots \geq d_{|V_1|}^{\text{bef}}$ and $d_1^{\text{aft}} \geq d_2^{\text{aft}} \geq \ldots \geq d_{|V_1|}^{\text{aft}}$ be the degree sequence (in the current semi-matching) before and after the removal of a cost-reducing path, then $\sum_{i=1}^{s} d_i^{\text{aft}} \leq \sum_{i=1}^{s} d_i^{\text{bef}}$ for any $1 \leq s \leq |V_1|$. Since this invariant is satisfied after every iteration of algorithm $\mathcal{A}_{SM2}$, it holds at the beginning and the end, and we have

$$\sum_{i=1}^{s} deg_{\mathcal{N}}(i) \leq \sum_{i=1}^{s} deg_{M'}(i) \tag{9}$$

Finally, observe that when we restrict M to only the set V_2', the sum of the top s-degrees can only *decrease*, i.e.

$$\sum_{i=1}^{s} deg_{M'}(i) \leq \sum_{i=1}^{s} deg_M(i) \tag{10}$$

Combining equations (9) and (10), the result follows. ■

Lemma 24 *Let $G = (V_1, V_2, E)$ be a bipartite graph and suppose that M is* any *semi-matching on G. Further, let $J = (V_1 \uplus V_3, V_2, E')$ be the supergraph of G obtained by adding nodes V_3 on the left and edges $E' \setminus E$ to G. Starting with M, we can use algorithm $\mathcal{A}_{SM2}$ in Harvey et al. (2003) to obtain an optimal semi-matching $\mathcal{N}$ for the supergraph J. Let the nodes in V_1 be indexed such that $deg_M(1) \geq deg_M(2) \geq \ldots \geq deg_M(|V_1|)$ and the nodes in $V_1 \uplus V_3$ indexed again such that $deg_{\mathcal{N}}(1) \geq deg_{\mathcal{N}}(2) \geq \ldots \geq deg_{\mathcal{N}}(|V_1 \uplus V_3|)$. Then for any $1 \leq s \leq |V_1 \uplus V_3|$, we have $\sum_{i=1}^{s} deg_{\mathcal{N}}(i) \leq \sum_{i=1}^{s} deg_M(i)$, i.e. the sum of the top s-degrees can only decrease as we go from M to $\mathcal{N}$.*

Proof Note that M is a feasible semi-matching for the graph J, with all nodes in V_3 having degree 0. We can repeat the argument from the previous lemma to show that if $d_1^{\text{bef}} \geq d_2^{\text{bef}} \geq \ldots \geq d_{|V_1 \cup V_3|}^{\text{bef}}$ and $d_1^{\text{aft}} \geq d_2^{\text{aft}} \geq \ldots \geq d_{|V_1 \cup V_3|}^{\text{aft}}$ be the degree sequence (in the current semi-matching) before and after the removal of any cost-reducing path, then $\sum_{i=1}^{s} d_i^{\text{aft}} \leq \sum_{i=1}^{s} d_i^{\text{bef}}$ for any $1 \leq s \leq |V_1 \cup V_3|$. The result then follows. ■

Notation for the proofs. Let $\mathcal{T}^+$ denote the set $\{t^+ : t \in \mathcal{T}\}$, and similarly $\mathcal{T}^-$ denote the set $\{t^- : t \in \mathcal{T}\}$, these are "task copies". Now partition the set of task copies $\mathcal{T}^+ \cup \mathcal{T}^-$ as $E \cup F$ such that for any task t, if the **true** label is $+1$, we put t^+ in E and t^- in F, otherwise, we put t^- in E and t^+ in F. Thus, E contains task copies with *true* labels while F contains task copies with *incorrect* labels. Let $F' \subseteq F$ denote the set of tasks for which honest workers provide incorrect responses, and denote the optimal semi-matching on the graph $\mathcal{B}_H^{cs}$ by $\mathcal{M}_H$. Without loss of generality, suppose that honest workers are indexed such that $d_1 \geq d_2 \geq \cdots \geq d_{|H|}$ where d_h denotes the degree of honest worker h in semi-matching $\mathcal{M}_H$.

Part 1. Adversary strategy that affects at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks

If $\varepsilon = 0$, there are no conflicts in the responses provided by honest workers, and therefore the bipartite graph $\mathcal{B}_H^{cs}$ contains just "true" task copies E on the right.

Given this, the adversaries *target* honest workers $\{1, 2, \ldots, k-1\}$: for each i, adversary a_i labels opposite to worker h_i (i.e. provides the incorrect response) on every task that h_i is mapped to in the semi-matching $\mathcal{M}_H$. Furthermore, the adversary uses its last identity a_k to provide the incorrect response on every task $t \in \mathcal{T}$ for which one of the first $k-1$ adversaries have not already labeled on. We argue that under the penalty-based aggregation algorithm, this adversary strategy results in incorrect labels for at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks. To see this, first note that the conflict set $\mathcal{T}_{cs} = \mathcal{T}$. Further, the bipartite graph $\mathcal{B}^{cs}$ decomposes into two disjoint subgraphs: bipartite graph $\mathcal{B}^{cs}(E)$ between H and E and semi-matching $M(F)$ between A and F that represents the adversary labeling strategy (it is a semi-matching because there is exactly one adversary that labels each task). Since the bipartite graph $\mathcal{B}^{cs}$ decomposes into two disjoint bipartite graphs, computing the optimal semi-matching on $\mathcal{B}^{cs}$ is equivalent to separately computing optimal semi-matchings on $\mathcal{B}^{cs}(E)$ and M_F. The argument above says that $\mathcal{B}^{cs}(E)$ is nothing but the graph $\mathcal{B}_H^{cs}$ defined in the theorem statement and therefore $\mathcal{M}_H$ is the optimal semi-matching on $\mathcal{B}^{cs}(E)$. Given that $M(F)$ is already a semi-matching by construction, the optimal semi-matching on $\mathcal{B}^{cs}$ is the disjoint union of $\mathcal{M}_H$ and $M(F)$. It is easy to see that in the resultant semi-matching, honest worker h_i and adversary a_i have the same degrees for $i = 1, 2, \ldots, k-1$. Hence, for every task mapped to honest worker h_i for $i = 1, 2, \ldots, k-1$ in the optimal semi-matching, the algorithm outputs a random label, and therefore outputs the correct label for at most half of these tasks. Thus, the above adversary strategy results in incorrect labels for at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks.

Part 2. Upper Bound on number of affected tasks

To simplify the exposition, we assume in the arguments below that the optimal semi-matching in the HARD PENALTY algorithm is computed for the entire task set and not just

the conflict set $\mathcal{T}_{cs}$. However, the bounds provided still hold as a result of lemma 23 above. Consequently, we abuse notation and use $\mathcal{B}^{cs}$ to denote the following bipartite graph in the remainder of the discussion: each worker w is represented by a node on the left, each task t is represented by at most two nodes on the right – t^+ and t^- – and we add an edge (w, t^+) if worker w labels task t as $+1$ and edge (w, t^-) if w labels t as -1. Then, it is easy to see that the subgraph of $\mathcal{B}^{cs}$ restricted to just the honest workers H on the left and task copies $E \cup F'$ on the right, is nothing but the graph $\mathcal{B}^{cs}_H$ defined in the theorem statement. Consequently, $\mathcal{M}_H$ is a feasible semi-matching for the subgraph $\mathcal{B}^{cs}(E \cup F')$, which is obtained by restricting $\mathcal{B}^{cs}$ to nodes $E \cup F'$ on the right. Further, we can assume that the adversary labeling strategy is always a semi-matching, i.e. there is at most one adversary response for any task. If the adversary labeling strategy is not a semi-matching, they can replace it with an alternate strategy where they only label for tasks to which they will be mapped in the optimal semi-matching (the adversaries can compute this since they have knowledge of the honest workers' responses). The optimal semi-matching doesn't change (otherwise it contradicts the optimality of the original semi-matching) and hence neither does the number of affected tasks.

We first state the following important lemma:

Lemma 25 *For any adversary labeling strategy, let $\mathcal{B}^{cs}(E \cup F')$ denote the bipartite graph $\mathcal{B}^{cs}$ restricted to all the workers $W = H \cup A$ on the left and, "true" task copies E as well as subset F' of the "incorrect" task copies F on the right. Let $\mathcal{M}$ be the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}$ and $\mathcal{M}(E \cup F') \subset M$ be the semi-matching $\mathcal{M}$ restricted to only the task copies $E \cup F'$. Then, $\mathcal{M}(E \cup F')$ is an optimal semi-matching for the sub-graph $\mathcal{B}^{cs}(E \cup F')$.*

Proof First observe that if $F' = F$, then the statement is trivially true. So we can assume $F' \subset F$. Suppose the statement is not true and let $\mathcal{N}(E \cup F')$ denote the optimal semi-matching on $\mathcal{B}^{cs}(E \cup F')$. We use $d_w(K)$ to denote the degree of worker w in a semi-matching K. Note that, $d_a(\mathcal{N}(E \cup F')) \leq d_a(\mathcal{M}(E \cup F')) \leq d_a(\mathcal{M})$ for all adversaries $a \in A$, where the first inequality follows from the fact that the adversary strategy is a semi-matching and the second inequality is true because $\mathcal{M}(E \cup F') \subset \mathcal{M}$. The adversaries who do not provide any responses for the task copies $E \cup F'$ will have degrees 0 in the semi-matchings $\mathcal{N}(E \cup F')$ and $\mathcal{M}(E \cup F')$ but the inequality is still satisfied. Now, since $\mathcal{N}(E \cup F')$ is an optimal semi-matching and $\mathcal{M}(E \cup F')$ is not, we have that

$$cost(\mathcal{N}(E \cup F')) < cost(\mathcal{M}(E \cup F')) \Rightarrow$$
$$\sum_{h \in H} d_h^2(\mathcal{N}(E \cup F')) + \sum_{a \in A} d_a^2(\mathcal{N}(E \cup F')) < \sum_{h \in H} d_h^2(\mathcal{M}(E \cup F')) + \sum_{a \in A} d_a^2(\mathcal{M}(E \cup F'))$$

Now, consider the semi-matching $\mathcal{N}$ on $\mathcal{B}^{cs}$ where we start with the semi-matching $\mathcal{N}(E \cup F')$ and then map the remaining task copies in $\mathcal{B}^{cs}$ (which belong to the set $F \setminus F'$) to the adversaries which they were assigned to in $\mathcal{M}$. We claim that $cost(\mathcal{N}) < cost(\mathcal{M})$ which is

a contradiction since $\mathcal{M}$ was assumed to be an optimal semi-matching on $\mathcal{B}^{cs}$. To see this:

$$
\begin{aligned}
&cost(\mathcal{M}) - cost(\mathcal{N}) \\
&= \sum_{h\in H} d_h^2(\mathcal{M}) + \sum_{a\in A} d_a^2(\mathcal{M}) - \left(\sum_{h\in H} d_h^2(\mathcal{N}) + \sum_{a\in A} d_a^2(\mathcal{N})\right) \\
&= \sum_{h\in H} d_h^2(\mathcal{M}(E\cup F')) + \sum_{a\in A} \left(d_a(\mathcal{M}(E\cup F')) + \Delta_a\right)^2 \\
&- \left(\sum_{h\in H} d_h^2(\mathcal{N}(E\cup F')) + \sum_{a\in A} \left(d_a(\mathcal{N}(E\cup F')) + \Delta_a\right)^2\right) \\
&\Big(\text{where } \Delta_a \stackrel{\text{def}}{=} d_a(\mathcal{M}) - d_a(\mathcal{M}(E\cup F')) \geq 0\Big) \\
&= \left(\sum_{h\in H} d_h^2(\mathcal{M}(E\cup F')) + \sum_{a\in A} d_a^2(\mathcal{M}(E\cup F')) - \sum_{h\in H} d_h^2(\mathcal{N}(E\cup F')) - \sum_{a\in A} d_a^2(\mathcal{N}(E\cup F'))\right) \\
&+ 2\sum_{a\in A} \left(d_a(\mathcal{M}(E\cup F')) - d_a(\mathcal{N}(E\cup F'))\right) * \Delta_a > 0 \\
&\big(\text{since } d_a(\mathcal{M}(E\cup F')) \geq d_a(\mathcal{N}(E\cup F')) \text{ as stated above}\big)
\end{aligned}
$$

Therefore, $\mathcal{M}(E\cup F')$ is an optimal semi-matching for the sub-graph $\mathcal{B}^{cs}(E\cup F')$. ■

Part 2a. Adversaries only provide incorrect responses or one adversary provides correct responses

Adversaries only provide incorrect responses. Let us begin with the case when all adversaries only provide incorrect responses.

Lemma 26 *Suppose that $\varepsilon = 0$ and adversaries only provide incorrect responses. Let M be an* arbitrary *semi-matching on the bipartite graph $\mathcal{B}^{cs}$ and suppose that this semi-matching is used in the* penalty-based aggregation *Algorithm to compute the true labels of the tasks. Further, let $b_1 \geq b_2 \geq \cdots \geq b_{|H|}$ denote the degrees of the honest workers in this semi-matching where b_i is the degree of honest worker h_i. Then, the number of affected tasks is at most $\sum_{i=1}^{k} b_i$.*

Proof It follows from the assumption that $\varepsilon = 0$ and that adversaries only provide incorrect responses, that there are no cross-edges between nodes H and F as well as A and E in the bipartite graph $\mathcal{B}^{cs}$. Thus, for any adversary labeling strategy, we can decompose $\mathcal{B}^{cs}$ into *disjoint* bipartite graphs $\mathcal{B}^{cs}(E)$ and $\mathcal{B}^{cs}(F)$, where $\mathcal{B}^{cs}(E)$ is the subgraph consisting of honest workers H and task copies E and $\mathcal{B}^{cs}(F)$ is the subgraph between the adversaries A and the task copies F. This further means that the semi-matching M is a disjoint union of semi-matchings on $\mathcal{B}^{cs}(E)$ and $\mathcal{B}^{cs}(F)$. Let the semi-matchings on the subgraphs be termed as $M(E)$ and $M(F)$ respectively. Further, let $T_{\text{aff}} \subseteq \mathcal{T}$ denote the set of tasks that are affected under this strategy of the adversaries and when the semi-matching M is used to compute the penalties of the workers in the penalty-based aggregation algorithm.

We claim that $|T_{\text{aff}}| \leq \sum_{i=1}^{k} b_i$. To see this, for each adversary $a \in A$, let $H(a) \subset H$ denote the set of honest workers who have "lost" to a i.e., for each worker $h \in H(a)$ there exists some task $t \in \mathcal{T}_{cs}$ such that h is mapped to the *true* copy of t in $M(E)$, a is mapped to the *incorrect* copy of t in $M(F)$, and the degree of h in $M(E)$ is greater than or equal to the degree of a in $M(F)$. Of course, $H(a)$ may be empty. Let $\bar{A}$ denote the set of adversaries $\{a \in A \colon H(a) \neq \emptyset\}$ and let $\bar{H}$ denote the set of honest workers $\bigcup_{a \in \bar{A}} H(a)$. Now define a bipartite graph between the nodes $\bar{A}$ and $\bar{H}$ with an edge between $a \in \bar{A}$ and $h \in \bar{H}$ if and only if $h \in H(a)$. This bipartite graph can be decomposed into degenerate and growth subgraphs by lemma 22 above. In the growth subgraph, by Hall's condition, we can find a perfect matching from adversaries to honest workers. Let (A_1, H_1) with $A_1 \subseteq \bar{A}$ and $H_1 = \text{Img}(A_1)$ be the degenerate component. The number of tasks that adversaries in A_1 affect is bounded above by $\sum_{h \in \text{Img}(A_1)} b_h$. Similarly, for $A_2 = \bar{A} \setminus A_1$, we can match each adversary to a *distinct* honest worker whose degree in $M(E)$ is greater than or equal to the degree of the adversary in $M(F)$. We can bound the number of affected tasks caused due to the adversaries in A_2 by the sum of their degrees, which in turn is bounded above by the sum of the degrees of honest workers that the adversaries are matched to. Let H_2 denote the set of honest workers matched to adversaries in the perfect matching. Thus, we have upper bounded the number of affected tasks by $\sum_{h \in H_1 \cup H_2} b_h$. It is easy to see that $|H_1 \cup H_2| \leq k$. Therefore, $\sum_{h \in H_1 \cup H_2} b_h \leq \sum_{i=1}^{k} b_i$. Therefore, the number of affected tasks $|T_{\text{aff}}|$ is at most $\sum_{i=1}^{k} b_i$. ■

We now extend the above lemma to the case when $\varepsilon \neq 0$.

Lemma 27 *Suppose that adversaries only provide incorrect responses. Let M be an* arbitrary *semi-matching on the bipartite graph $\mathcal{B}^{cs}$ and suppose that this semi-matching is used in the* PENALTY-BASED AGGREGATION *Algorithm to compute the true labels of the tasks. Further, let $b_1 \geq b_2 \geq \cdots \geq b_{|H|}$ denote the degrees of the honest workers in this semi-matching where b_i is the degree of honest worker h_i. Then, the number of affected tasks is at most $\sum_{i=1}^{k+\varepsilon \cdot |H|} b_i$.*

Proof Since honest workers can make mistakes, there are cross-edges between H and F in the bipartite graph $\mathcal{B}^{cs}$. Observe that there are two kinds of affected tasks in this case: (1) tasks that adversaries "win" against honest workers, say $T_{\text{aff}}(A)$, similar to those in the previous lemma and (2) tasks that are affected when 2 honest workers are compared in the final step of the PENALTY-BASED AGGREGATION algorithm, say $T_{\text{aff}}(H)$. We can repeat the argument from lemma 26 above to bound $|T_{\text{aff}}(A)|$ by the sum of the degrees of (some) k honest workers, say H_k, in semi-matching M. Further, we can bound $|T_{\text{aff}}(H)|$ by the sum of the degrees of honest workers who make mistakes, in the semi-matching M. By assumption, there are at most $\varepsilon \cdot |H|$ such workers. Now if any of these workers belong to H_k, then we have already accounted for their degree. Consequently, we have that $|T_{\text{aff}}(H)| + |T_{\text{aff}}(A)|$ is upper bounded by the sum of the degrees of the top $k + \varepsilon \cdot |H|$ honest workers in the semi-matching M, which establishes the result. ■

Since the above lemmas are true for *any* choice of semi-matching M, they hold in particular for the optimal semi-matching on $\mathcal{B}^{cs}$. Therefore, it gives us an upper bound on the number of affected tasks when the adversaries only provide incorrect responses.

One adversary provides correct responses. Next consider the case when there is exactly 1 adversary that provides correct responses and all other adversaries only provide incorrect responses. Let $\mathcal{M}$ be the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}$ resulting from such an adversary strategy and let $\bar{a}$ denote the adversary who provides correct responses. Observe that we can repeat the argument from lemma 27 above to get an upper bound on the number of affected tasks that the adversaries "win" against honest workers as well as those that honest workers who make mistakes "win" against other honest workers. Let T_1 be the collection of such tasks. In the proof of lemma 26, there are two possible scenarios: either we obtain a perfect matching between the k adversaries and some k honest workers in which case we have accounted for all of the affected tasks that come from adversaries winning (against honest workers or $\bar{a}$). In the other scenario, when the degenerate component is non-empty, we have at most $k-1$ honest workers on the right and we bound the number of tasks that adversaries "win" by the sum of the degrees of these honest workers. Note however that we may be missing out on some of the affected tasks, namely those that the adversary $\bar{a}$ "loses" against *other adversaries* (the losses against honest workers who make mistakes are already accounted for). The tasks that we might be missing out on correspond exactly to the task copies in E that the adversary $\bar{a}$ is mapped to in the optimal semi-matching $\mathcal{M}$.

Next observe that in both scenarios above—all adversaries provide incorrect responses and exactly one adversary provides correct responses—we can upper bound the total number of affected tasks by the sum of the degrees of some $k + \varepsilon \cdot |H|$ workers in the optimal semi-matching $\mathcal{M}$ restricted to just the task copies $E \cup F'$ on the right (since honest workers provide responses only on these tasks), which we denote as $\mathcal{M}(E \cup F')$. Lemma 25 tells us that $\mathcal{M}(E \cup F')$ is, in fact, the optimal semi-matching on the subgraph $\mathcal{B}^{cs}(E \cup F')$ between workers W and the task copies $E \cup F'$. In addition, lemma 24 tells us that this sum is at most $\sum_{i=1}^{k+\varepsilon\cdot|H|} d_i$ (by starting with $\mathcal{M}_H$ as a feasible semi-matching) and the bound follows.

Part 2b. Adversaries can provide arbitrary responses

Consider the general case when all adversaries can provide arbitrary responses. First recall that lemma 27 was applicable to any semi-matching and in fact, we can use the argument even when adversaries provide correct responses. Formally, consider an arbitrary adversary strategy resulting in an optimal semi-matching $\mathcal{M}$ on $\mathcal{B}^{cs}$. Let $\mathcal{M}(E \cup F')$ denote the semi-matching M restricted to just the task copies $E \cup F'$. Suppose that the set of affected tasks T_{aff} under this adversary strategy is such that $T_{\text{aff}} = T_{\text{aff}}(A_1) \cup T_{\text{aff}}(H) \cup T_{\text{aff}}(A_2)$ where $T_{\text{aff}}(A_1)$ are the tasks that the adversaries "win" against honest workers, $T_{\text{aff}}(H)$ are the tasks that honest workers who make mistakes "win" (against other honest workers and/or adversaries) and $T_{\text{aff}}(A_2)$ are the tasks that are affected when 2 adversaries are compared against each other in the final step of the penalty-based aggregation algorithm. We can then utilize the argument in lemma 27 to bound $|T_{\text{aff}}(H)| + |T_{\text{aff}}(A_1)|$ by the sum of the degrees of (some) $k + \varepsilon \cdot |H|$ honest workers in the optimal semi-matching $\mathcal{M}$ (the

tasks affected when honest workers who make mistakes win against adversaries are also accounted). Further, we can bound $|T_{\text{aff}}(A_2)|$ by the sum of the degrees of the adversaries in the semi-matching $\mathcal{M}(E)$, which is the semi-matching $\mathcal{M}$ restricted to task copies E.

Let $A(H) \subseteq A$ denote the adversaries that have non-zero degrees in semi-matching $\mathcal{M}(E)$, i.e. they are mapped to some task copy in E in semi-matching $\mathcal{M}$. The above sequence of claims implies that we can bound the number of affected tasks $|T_{\text{aff}}|$ by the sum of the degrees of the top $s = k + \varepsilon \cdot |H| + |A(H)|$ workers in the semi-matching $\mathcal{M}(E \cup F')$, which is $\mathcal{M}$ restricted to the task copies $E \cup F'$ (because honest workers, by assumption, only provide responses on task copies $E \cup F'$ and the semi-matching $\mathcal{M}(E) \subseteq \mathcal{M}(E \cup F')$). Now, we claim that this itself is upper bounded by the sum of the degrees of the top s honest workers in the optimal semi-matching $\mathcal{M}_H$ on the bipartite graph $\mathcal{B}_H^{cs}$. To see this, start with $\mathcal{M}_H$ as a feasible semi-matching from workers W to task copies $E \cup F'$ (recall that it is feasible since we assume that each task has at least one correct response from honest workers). Then, lemma 24 tells us that the sum of the degrees of the top s workers in the optimal semi-matching on $\mathcal{B}^{cs}(E \cup F')$ is at most the sum of the degrees of the top s honest workers in $\mathcal{M}_H$. Further, lemma 25 tells us that the optimal semi-matching on the subgraph $\mathcal{B}^{cs}(E \cup F')$ is precisely the semi-matching $\mathcal{M}(E \cup F')$. This shows that we can bound the number of affected tasks by $\sum_{i=1}^{s} d_i$. Finally, note that $|A(H)| \leq k \Rightarrow s \leq 2k + \varepsilon \cdot |H|$ and hence, we can bound the total number of affected tasks by $\sum_{i=1}^{2k+\varepsilon \cdot |H|} d_i$.

A.7 Uniqueness of degree-sequence in optimal semi-matchings

In the arguments above, we have implicitly assumed some sort of uniqueness for the optimal semi-matching on any bipartite graph. Clearly its possible to have multiple optimal semi-matchings for a given bipartite graph. However, we prove below that the degree sequence of the vertices is unique across all optimal semi-matchings and hence our bounds still hold without ambiguity.

Lemma 28 *Let M and M' be two optimal semi-matchings on a bipartite graph $G = (V_1, V_2, E)$ with $|V_1| = n$ and let $d_1 \geq d_2 \cdots \geq d_n$ and $d'_1 \geq d'_2 \geq \cdots d'_n$ be the degree sequence for the V_1-vertices in M and M' respectively. Then, $d_i = d'_i \; \forall \; 1 \leq i \leq n$, or in other words, any two optimal semi-matchings have the same degree sequence.*

Proof Let s be the smallest index such that $d_s \neq d'_s$, note that we must have $s < n$ since we have that $\sum_{j=1}^{n} d'_j = \sum_{j=1}^{n} d_j$. This means that we have $d_j = d'_j \; \forall j < s$. Without loss of generality, assume that $d'_s > d_s$. Now, $\exists q \in \mathbb{N}$ such that $d'^q_s > d^q_s + \sum_{j=s+1}^{n} d^q_j$ and since $d_j = d'_j \; \forall j < s$, we have that $\sum_{j=1}^{n} d'^q_j \geq \sum_{j=1}^{l} d'^q_j > \sum_{j=1}^{n} d^q_j$. But, this is a contradiction since an optimal semi-matching minimizes the $\mathcal{L}_p$ norm of the degree-vector of V_1-vertices for any $p \geq 1$ (Section 3.4 in Harvey et al. (2003)). Hence, we have that $d_i = d'_i \; \forall i$. ■

A.8 Relation between Lower bound L and Optimal semi-matching degrees

We prove here the relationship between the lower bound L in theorem 14 and the honest worker degrees $d_1, d_2, \ldots, d_{|H|}$ in the optimal semi-matching on the bipartite graph $\mathcal{B}_H^{cs}$.

Lemma 29 *Suppose that honest workers are perfectly reliable and let $d_1 > d_2 > \cdots > d_{|H|}$ denote the degrees of the honest workers in the optimal semi-matching $\mathcal{M}_H$ on $\mathcal{B}_H^{cs}$. Then the lower bound L in theorem 14 is such that $L \geq \sum_{i=1}^{k-1} d_i$.*

Proof Let $T_1, T_2, \ldots, T_{k-1}$ denote the set of tasks that are mapped to honest workers $h_1, h_2, \ldots, h_{k-1}$ in $\mathcal{M}_H$ and $T := \bigcup_{j=1}^{k-1} T_j$. Now, we claim that for any $t \in T$, the only honest workers that provide responses for t are amongst $h_1, h_2, \ldots, h_k$. In other words, $\text{PreIm}(T) \subseteq \{h_1, h_2, \ldots, h_k\}$. Suppose not, so that there exists $h_i \in \text{PreIm}(T)$ such that $i > k$. This would contradict the fact that $\mathcal{M}_H$ is an optimal semi-matching. Specifically, Theorem 3.1 in Harvey et al. (2003) shows that a semi-matching M is optimal if and only if there is no *cost-reducing path* relative to M. A cost-reducing path $P = (h^{(1)}, t^{(1)}, h^{(2)}, \ldots h^{(d)})$ for a semi-matching M on $\mathcal{B}_H^{cs}$ is an alternating sequence of honest workers and tasks such that $t^{(x)}$ is mapped to $h^{(x)}$ in the semi-matching M for all $1 \leq x \leq d-1$ and $deg_M(h^{(1)}) > deg_M(h^{(d)}) + 1$. Here $deg_M()$ denotes the degree of a node in the semi-matching M (see section 2.1 in Harvey et al. (2003) for a precise definition). Since $i > k$, we have that $d_s > d_i + 1$ for all $s \in \{1, 2, \ldots, k-1\}$, which introduces a cost-reducing path. Therefore, we have that $\text{PreIm}(T) \subseteq \{h_1, h_2, \ldots, h_k\}$. Now

$$|T| = |\bigcup_{j=1}^{k-1} T_j| = \sum_{j=1}^{k-1} |T_j| = \sum_{j=1}^{k-1} d_j$$

where we have used the property of a semi-matching that a given task is mapped to only one worker. Using the definition of the lower bound L, it follows that $L \geq |T| = \sum_{i=1}^{k-1} d_i$. ■

Appendix B. Experimental Details

In this section we describe the details of our experimental setup discussed in section 5. We start with the benchmark algorithms.

EM algorithm. We consider the EM algorithm proposed by Raykar and Yu (2012). The worker model they consider is as follows: for each worker w_i, her accuracy is modeled separately for positive and negative tasks (referred to as the "two-coin" model). For a task t_j with true label $+1$, the *sensitivity* (true positive rate) for worker w_i is defined as:

$$\alpha_i := \Pr[w_i(t_j) = +1 \,\Big|\, y_j = +1]$$

Similarly, the *specificity* (1- false positive rate) is defined as:

$$\beta_i := \Pr[w_i(t_j) = -1 \,\Big|\, y_j = -1]$$

Let $\mathbf{\Theta} = [\Big\{(\alpha_i, \beta_i) \,\Big|\, i \in [n]\Big\}, \gamma]$ denote the set of all parameters. For ease of exposition, we assume that the worker-task assignment graph is complete, i.e. all workers provide responses for all items, but the algorithm can be immediately extended to the case of

incomplete graphs. Given the response matrix $\mathcal{L}$, the log-likelihood of the parameters Θ can be written as:

$$\log Pr[\mathcal{L} \Big| \mathbf{\Theta}] = \sum_{j=1}^{m} \log \left(\prod_{i=1}^{n} {\alpha_i}^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (1-\alpha_i)^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \cdot \gamma + \prod_{i=1}^{n} (1-\beta_i)^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot {\beta_i}^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \cdot (1-\gamma) \right)$$

The MLE of the parameters can be computed by introducing the latent true label of each task, denoted by the vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$. The complete data log-likelihood can then be written as:

$$\log Pr[\mathcal{L}, \boldsymbol{y} \Big| \mathbf{\Theta}] = \sum_{j=1}^{m} \left(y_j \log(a_j \gamma) + (1-y_j) \log(1-\gamma) b_j \right) \tag{11}$$

where

$$a_j = \prod_{i=1}^{n} {\alpha_i}^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (1-\alpha_i)^{\mathbf{1}[\mathcal{L}_{ij}=-1]}$$
$$b_j = \prod_{i=1}^{n} (1-\beta_i)^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot {\beta_i}^{\mathbf{1}[\mathcal{L}_{ij}=-1]}$$

Each iteration of the EM algorithm consists of two steps:

- **E-step:** Given the response matrix $\mathcal{L}$ and the current estimate of the model parameters $\mathbf{\Theta}^{(k)}$, the conditional expectation of the complete data log-likelihood is computed as

  $$\mathbb{E}\left\{ \log \Pr[\mathcal{L}, \boldsymbol{y} \Big| \mathbf{\Theta}^{(k)}] \right\} = \sum_{j=1}^{m} \left(\gamma_j^{(k)} \log(a_j^{(k)} \gamma^{(k)}) + (1-\gamma_j^{(k)}) \log(1-\gamma^{(k)}) b_j^{(k)} \right)$$

  where the expectation is w.r.t to $\Pr[\boldsymbol{y} \Big| \mathcal{L}; \mathbf{\Theta}^{(k)}]$ and $\gamma_j^{(k)} = \Pr[y_j = +1 \Big| \mathcal{L}; \mathbf{\Theta}^{(k)}]$. Using Bayes theorem, we can compute

  $$\gamma_j^{(k)} \propto \Pr[\mathcal{L}_{1j}, \mathcal{L}_{2j}, \ldots, \mathcal{L}_{nj} \Big| y_j = +1; \mathbf{\Theta}^{(k)}] \cdot \Pr[y_j = +1 \Big| \mathbf{\Theta}^{(k)}] = \frac{a_j^{(k)} \gamma^{(k)}}{a_j^{(k)} \gamma^{(k)} + b_j^{(k)} (1-\gamma^{(k)})}$$

- **M-step:** Based on the current posterior estimates of the true labels $\gamma_j^{(k)}$ and the response matrix $\mathcal{L}$, the model parameters are updated by maximizing $\mathbb{E}\left\{ \log \Pr[\mathcal{L}, \boldsymbol{y} \Big| \mathbf{\Theta}^{(k)}] \right\}$, which can be shown to be a lower bound on the data log-likelihood (eq 11). The prevalence of positive tasks γ is updated as:

  $$\gamma^{(k+1)} = \frac{\sum_{j=1}^{m} \gamma_j^{(k)}}{m}$$

Similarly, the parameters α_i, β_i are updated as:

$$\alpha_i^{(k+1)} = \frac{\sum_{j=1}^m \mathbf{1}[\mathcal{L}_{ij} = +1]\gamma_j^{(k)}}{\sum_{j=1}^m \gamma_j^{(k)}}$$

$$\beta_i^{(k+1)} = \frac{\sum_{j=1}^m \mathbf{1}[\mathcal{L}_{ij} = -1](1-\gamma_j^{(k)})}{\sum_{j=1}^m (1-\gamma_j^{(k)})}$$

These two steps are iterated until convergence of the log-likelihood $\Pr[\mathcal{L} \,\big|\, \mathbf{\Theta}]$. To initialize the EM algorithm, we use the majority estimate $\gamma_j^{(0)} = \frac{\sum_{i=1}^n \mathbf{1}[\mathcal{L}_{ij}=+1]}{n}$.

We implement the above algorithm in our experiments. Note that the model we consider in the theoretical analysis is the simpler "one-coin" model where every worker is characterized by only a single parameter μ_i - the probability that she labels an assigned task correctly. The EM algorithm for that case can be derived in a similar manner to the one described above; see the proof of Lemma 13 above.

KOS algorithms. We implemented the iterative algorithm presented in Karger et al. (2014) which we replicate below in our notation.

Algorithm 4 KOS algorithm

1: **Input:** $\mathcal{L}$,$\mathcal{B} = (W, \mathcal{T}, E)$, $k_{\max}$.
2: For all $(w_i, t_j) \in E$, initialize $m_{i\to j}^{(0)}$ with random $Z_{ij} \sim \mathcal{N}(1,1)$
3: For $k = 1, 2, \ldots, k_{\max}$,

- For all $(w_i, t_j) \in E$, update $m_{j\to i}^{(k)} = \sum_{i' \neq i} \mathcal{L}_{ij} m_{i\to j}^{(k-1)}$
- For all $(w_i, t_j) \in E$, update $m_{i\to j}^{(k)} = \sum_{j' \neq j} \mathcal{L}_{ij} m_{j\to i}^{(k)}$

4: For all t_j, compute $m_j = \sum_{i=1}^n \mathcal{L}_{ij} m_{i\to j}^{(k_{\max}-1)}$
5: **Output:** label for task t_j as $\hat{y}_j = \text{sign}(\text{m}_\text{j})$

This algorithm was proposed for random regular graphs in the paper and we modified it in the following way for use in non-regular graphs:

Algorithm 5 KOS(NORM) algorithm

1: **Input:** $\mathcal{L}$,$\mathcal{B} = (W, \mathcal{T}, E)$, $k_{\max}$.
2: For all $(w_i, t_j) \in E$, initialize $m_{i\to j}^{(0)}$ with random $Z_{ij} \sim \mathcal{N}(1,1)$
3: For $k = 1, 2, \ldots, k_{\max}$,

- For all $(w_i, t_j) \in E$, update $m_{j\to i}^{(k)} = \frac{1}{deg_{\mathcal{B}}(t_j)} \sum_{i' \neq i} \mathcal{L}_{ij} m_{i\to j}^{(k-1)}$
- For all $(w_i, t_j) \in E$, update $m_{i\to j}^{(k)} = \frac{1}{deg_{\mathcal{B}}(w_i)} \sum_{j' \neq j} \mathcal{L}_{ij} m_{j\to i}^{(k)}$

4: For all t_j, compute $m_j = \sum_{i=1}^n \mathcal{L}_{ij} m_{i\to j}^{(k_{\max}-1)}$
5: **Output:** label for task t_j as $\hat{y}_j = \text{sign}(\text{m}_\text{j})$

We chose $k_{\max} = 100$ in our experiments.

Simulation Details. Here we discuss how we imposed the degree bias on adversaries in our simulation study. Given a worker-task assignment graph, let d_w denote the degree of worker w and $d_{\min}, d_{\text{avg}}, d_{\max}$ denote resp. the minimum, average and maximum worker degrees.

- *Adversaries have high degrees.* For each worker w, define $q_w = q \cdot \frac{d_{\max} - d_w}{d_{\max} - d_{\text{avg}}}$. Then, each worker w is an adversary with probability $1 - q_w$. First, note that the expected number of honest workers is given by

$$\sum_w q_w = \frac{q}{d_{\max} - d_{\text{avg}}} \sum_w (d_{\max} - d_w) = q \cdot n$$

  Next, we can see that workers with higher degrees have a smaller q_w, which implies that they have a greater chance of being an adversary. In an analogous manner, we deal with the case of low degrees.

- *Adversaries have low degrees.* For each worker w, define $q_w = q \cdot \frac{d_w - d_{\min}}{d_{\text{avg}} - d_{\min}}$. Then, each worker w is an adversary with probability $1 - q_w$. Again, we have that the expected number of honest workers is given by

$$\sum_w q_w = \frac{q}{d_{\text{avg}} - d_{\min}} \sum_w (d_w - d_{\min}) = q \cdot n$$

  In this case, lower the degree d_w, higher the chance $(1 - q_w)$ that a worker is chosen as an adversary.

References

Béla Bollobás. *Random Graphs*. Cambridge University Press, 2001.

Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 64–72, 2013.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 285–294. ACM, 2013.

George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proceedings of the Network and Distributed System Security Symposium*, 2009.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28, 1979.

Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *Proceedings of the First International Workshop on Crowdsourcing Web Search*, pages 26–30, 2012.

Julie S Downs, Mandy B Holbrook, Steve Sheng, and Lorrie Faith Cranor. Are your participants gaming the system?: Screening mechanical turk workers. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, pages 2399–2402. ACM, 2010.

Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: Answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, pages 61–72. ACM, 2011.

Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1631–1640. ACM, 2015.

Chao Gao and Dengyong Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *arXiv preprint arXiv:1310.5764v6*, 2016.

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. ACM, 2011.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.

Nicholas JA Harvey, Richard E Ladner, László Lovász, and Tami Tamir. Semi-matchings for bipartite graphs and load balancing. In *Algorithms and Data Structures*, pages 294–306. Springer, 2003.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. Learning whom to trust with mace. In *Proceedings of NAACL-HLT*, pages 1120–1130, 2013.

Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67. ACM, 2010.

Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 2492–2500, 2014.

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

David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems*, pages 1953–1961, 2011.

David R Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. In *Proceedings of ACM SIGMETRICS*, pages 81–92. ACM, 2013.

David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

Ashish Khetan and Sewoong Oh. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 4844–4852, 2016.

Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–456. ACM, 2008.

John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Search Evaluation*, 2010.

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

Hongwei Li, Bo Zhao, and Ariel Fuxman. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 165–176. ACM, 2014.

Qiang Liu, Jian Peng, and Alex Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 701–709, 2012.

Arash Molavi Kakhki, Chloe Kliman-Silver, and Alan Mislove. Iolaus: Securing online content rating systems. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 919–930. ACM, 2013.

Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M Voelker. Dirty jobs: The role of freelance labor in web service abuse. In *Proceedings of the 20th USENIX conference on Security*, pages 14–29. USENIX Association, 2011.

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.

Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web*, pages 191–200. ACM, 2012.

Jungseul Ok, Sewoong Oh, Jinwoo Shin, and Yung Yi. Optimality of belief propagation for crowdsourced classification. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*, pages 535–544, 2016.

Aditya Ramesh, Aditya Parameswaran, Hector Garcia-Molina, and Neoklis Polyzotis. Identifying reliable workers swiftly. 2012.

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 Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 614–622. ACM, 2008.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems*, pages 1085–1092, 1995.

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

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

Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, pages 15–28. USENIX Association, 2009.

Nguyen Tran, Jinyang Li, Lakshminarayanan Subramanian, and Sherman SM Chow. Optimal sybil-resilient node admission control. In *Proceedings of IEEE INFOCOM*, pages 3218–3226. IEEE, 2011.

Salil P Vadhan et al. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 40:363–374, 2010.

Bimal Viswanath, Mainack Mondal, Krishna P Gummadi, Alan Mislove, and Ansley Post. Canal: Scaling social network-based sybil tolerance schemes. In *Proceedings of the 7th ACM European Conference on Computer Systems*, pages 309–322. ACM, 2012.

Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR11)*, pages 21–26, 2011.

Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. Serf and turf: Crowdturfing for fun and profit. In *Proceedings of the 21st International Conference on World Wide Web*, pages 679–688. ACM, 2012.

Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *23rd USENIX Security Symposium*. USENIX Association, 2014.

Naiyan Wang and Dit-Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1107–1115, 2014.

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

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

Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: Defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36:267–278, 2006.

Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17, 2008.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 1260–1268, 2014.

Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012.

Dengyong Zhou, Qiang Liu, John C Platt, Christopher Meek, and Nihar B Shah. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*, 2015.