Journal of Machine Learning Research 1 (2016) 1-58 Submitted 7/16; Published

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:

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 work assumes that worker responses follow specific probabilistic models, but recent evidence shows the presence of workers adopting non-random or even malicious strategies. To account for such workers, we suppose that the workers are comprised of a mixture of honest and adversarial workers. *Honest* workers may be reliable or unreliable, and 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 majority and 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 accuracies of several state-of-the-art label aggregation algorithms on real-world crowdsourcing datasets.

Keywords: Crowdsourcing, Reputation, Adversary, Outliers

1. Introduction

The growing popularity of online crowdsourcing services like Amazon Mechanical Turk, CrowdFlower etc. has made it easy to collect low-cost labels from the crowd to generate training datasets for machine learning applications. Unfortunately, the collected labels typically are of low quality because of unintentional or intentional inaccuracies introduced by unreliable and malicious workers (Kittur et al., 2008; Le et al., 2010). Determining the correct labels of the tasks from such noisy labels is challenging because the reliabilities or qualities of workers are often unknown. While one may use "gold standard" tasks – tasks whose true label is already known – to identify the low reliability workers (Snow et al., 2008; Downs et al., 2010; Le et al., 2010), getting access to the true labels for a sufficient number of tasks can be hard and expensive. To address these challenges, the common solution is to use *redundancy* (Sheng et al., 2008): collect multiple labels for each task and assign multiple tasks to each worker. Given the redundant labels, most existing work makes specific probabilistic assumptions on how individual workers provide labels and proposes techniques to identify low reliability workers, and either filter them out or de-emphasize their contribution. Common probabilistic models include the one-coin model (Zhang et al., 2014), the 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 work (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 the case of 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 the true labels in the corpus of tasks [1]; (b) provide accurate labels to the first few tasks but random labels to the remaining tasks; (c) systematically provide $+1$ labels to certain types of tasks and -1 to other types. See recent work by Vuurens et al. (2011) for real-world empirical evidence of similar worker strategies. In addition, workers may be malicious and adopt sophisticated strategies with the explicit purpose of altering the inferred labels for the tasks. For instance, Wang et al. (2014) show that malicious crowdsourcing campaigns, also called "crowdturfing", are growing in popularity in dedicated as well as generic crowdsourcing websites (Motoyama et al., 2011; Wang et al., 2012). Further, there is evidence for the presence of malicious users in online content rating systems (such as Digg, Amazon.com, Yelp.com, etc.), in which users can choose which 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.com (Mukherjee et al., 2012).

Motivated by the presence of non-random worker strategies, we go beyond standard probabilistic models and study the problem of inferring the task labels under a much broader class of *adversarial* worker strategies. We distinguish between two types of workers: *honest* and *adversarial*. The worker types are latent and our objective is to use only the labels provided by the workers as input to identify the adversaries. We consider a general crowdsourcing setting in which users/workers provide labels to items/tasks. The setting maybe a crowdsourced classification application (such as Mechanical Turk) in which labels are collected for tasks by assigning [2] them to workers or a public crowdsourced system (such as Digg, Amazon, 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. 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 the assigned tasks, but the assignment can be done in order to ensure that the graph denoting the assignment of workers to tasks has particular structures; see Section 4.1.

$\{-1, +1\}$; for cases in which the notion of task true label is subjective, we consider it to be the majority opinion of the population. The population of workers is mostly honest, with adversaries comprising a "small" fraction. The honest workers adopt a well-defined probabilistic labeling strategy (e.g. the one-coin model introduced above), so they *can make mistakes*. However, we make no assumptions on the specifics of the probabilistic strategy. The adversaries adopt strategies different from that of the honest workers, whether probabilistic or not. Further, different adversaries may adopt distinct strategies.

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. Note that we make no specific assumptions on the honest or adversary labeling strategies. Therefore, we base our algorithms on the intuition that as the population is mostly honest and the 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. We show in Section 5 that discarding the adversary labels can enable standard label aggregation algorithms to infer task true labels more accurately.

1.1 Main contributions

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

Algorithmic contributions. Our main algorithmic contribution is a reputation algorithm, designed to identify outlier labeling patterns. The algorithm takes as input the set of workers, the set of tasks, and the 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* the workers for the number of 'conflicts' they are involved in. More precisely, suppose every task receives both $+1$ and -1 labels. For each task t_j, the algorithm maintains the number d_j^+ of $+1$ labels, the 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 $+1$. Based on this intuition, 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 on how they make these two decisions. The first algorithm (Algorithm 1) does a 'soft' assignment: 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 allocated penalties across all tasks assigned to a worker [4]. The second algorithm (Algorithm 2) relies on 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; and the net penalty for the worker is computed by summing the allocated penalties. If the representative worker

3. This work expands on the results described in a previous version (Jagabathula et al., 2014)
4. We do not explicitly define the reputation score, it can be interpreted as the inverse of the net penalty – higher the penalty, lower the reputation and vice-versa.

is chosen uniformly at random, then a worker who agrees with the majority is less likely to be chosen and thereby 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 an *optimal semi-matching* (Harvey et al., 2003), where the semi-matching is defined on the bipartite graph between the workers and tasks – where every worker is connected to the tasks that she has labeled. Both penalty assignments are based on the intuition that when there are more honest workers than adversaries, the honest workers are more likely to agree with the majority and thereby receive lower penalties.

Theoretical contributions. We analyze our algorithms under three settings: (a) there are no adversaries; (b) the adversaries adopt the Uniform strategy, in which they label all assigned tasks +1; and (c) the adversaries are sophisticated, having infinite computational capacity and knowledge of the honest worker 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 such an 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 all the adversaries as honest.

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, i.e., each worker labels l tasks and each task is labeled by r workers; and the population is described by three parameters: the fraction q of honest workers, the fraction γ of tasks with +1 true labels, and the average reliability $\mu > 1/2$ of honest workers. 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 a crowdsourced classification task, in which the tasks may be assigned according to an (l, r)-regular graph. The (l, r)-regular graph is analytically tractable and has 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 budget for task assignment. We derive the expected penalties received by honest and adversarial workers and bound the asymptotic *error rate* or the *misclassification rate* of a threshold classifier – given a penalty threshold θ, the threshold classifier classifies a worker as honest if her penalty is less than or equal to θ and as adversarial, otherwise. The misclassification rate, then, is defined as the expected fraction of errors made by this classifier.

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.

More 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 – the higher the reliability, the lower the penalty; see Theorem 2. When $l = \log n$ and r is fixed, 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 she has high reliability (above a threshold) but classified as adversarial; see Theorem 5. 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 go 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 the adversaries; see Theorem 3. Our conditions essentially require a sufficient majority for the honest workers; more precisely, 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 3. This result shows that as γ increases, it becomes harder to separate the honest workers from the adversaries. The reason is that as the proportion γ of tasks with true labels $+1$ increases, the adversaries (who always provide the label $+1$) become more accurate, making them harder to separate from the honest workers. Further, when $l = \log n$ and r is fixed, we show that the misclassification rate (when the threshold classifier incorrectly classifies a worker as honest/adversarial) scales as $O(1/n^{2\varepsilon^2})$ for the 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 9. Here, $F(\cdot)$ is the cumulative distribution function (CDF) of the honest worker reliabilities. Our result essentially shows that asymptotically, we correctly classify all the adversaries and honest workers with above average reliabilities. We argue that the presence of the term $F(\tilde{\mu})$ is necessary because it is hard to separate low-reliable honest workers from adversaries.

3. *Sophisticated adversaries.* We suppose that the goal of the sophisticated adversaries is to maximize the number of tasks they *affect*, i.e., cause to receive labels different from what they would have received in their absence. When the honest workers are perfectly reliable, we 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 employed to aggregate the worker labels (as long as it is agnostic to worker/task identities). The bound depends on the graph structure between the honest workers and the tasks (see Theorem 12 for details). Our result is valid across different labeling patterns and a large class of label aggregation algorithms, and hence provides *fundamental* limits on the damage that k adversaries can cause. Further, we propose a label aggregation algorithm utilizing the worker reputations computed by the hard-penalty algorithm (Algorithm 2) and prove the existence of an upper bound on the worst-case number of affected tasks (see Theorem 15). This combined with the result of Theorem 12 shows that our proposed label aggregation algorithm is optimal (up to a constant factor) in recovering the true labels of the 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. On five real-world crowdsourcing data sets, it shows that *discarding* the labels of the adversaries identified by our methods allows standard label aggregation algorithms to infer task true labels more accurately. It demonstrates these improvements for the following four label aggregation algorithms: (a) simple majority, (b) EM (Raykar and Yu, 2012), (c) KOS (Karger et al., 2011), and (d) a normalized variant of KOS in which the messages are scaled by the corresponding worker and task degrees [5]. Our results show that by removing up to 10 workers, our methods can result in up to 30% improvement in the predictive accuracy. These improvements suggest that the label patterns of the discarded workers do not conform to standard probabilistic models.

The second study is designed to complement our theoretical analysis. Using synthetic data, we show that both the soft-penalty and hard-penalty algorithms successfully identify Uniform adversaries (who label all the assigned tasks +1) and low-reliable 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 tasks to label, with some workers labeling many tasks and some tasks receiving many labels (Franklin et al., 2011). The study also offers insights into the settings under which the soft- or the hard-penalty algorithm is appropriate: 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.

1.2 Related Work

Our work is part of the literature in crowdsourcing that proposes statistical techniques to exploit the redundancy in the collected labels to simultaneously infer the latent reliabilities of workers and the true labels of tasks. In particular, our work is related to three broad streams. The first stream focuses on 'crowdsourced classification' – inferring the underlying true labels of the 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 of work proposes methods to explicitly filter out low-reliability workers and is similar in spirit to our approach. Finally, the third stream focuses on methods to address sophisticated attacks in online settings and is related to our treatment of sophisticated adversaries.

Crowdsourced classification. The literature on crowdsourced classification is vast. Most of these works 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 the worker reliability parameters and task true 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) include 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) introduce a natural generalization of the Dawid-Skene model that captures tasks with differing difficulties, and propose 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

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.

no theoretical guarantees on the resulting estimates of the task true labels and the model parameters.

From the 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 the number of workers) grows large, each task receives an increasing number of responses, so that eventually the true labels of all the tasks are 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 the task labels. More recently, Gao and Zhou (2013) study the minimax optimal error rate of a projected EM algorithm under the one-coin model and, Li and Yu (2014) provide upper bounds on the error rate of weighted majority voting algorithms for the Dawid-Skene model. Zhang et al. (2014) show that the EM algorithm for the Dawid-Skene model achieves the optimal convergence rate when initialized using a spectral method.

The *sparse* regime, on the other hand, deals with the case when each task is assigned a "small" number of workers, i.e, 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 and proposed an iterative message-passing algorithm for estimating the 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 show that their algorithm is optimal by comparing against an oracle estimator who 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. Very recently, Khetan and Oh (2016) analyzed the "Generalized Dawid-Skene Model" introduced by Zhou et al. and show that spectral approaches achieve near-optimal performance, whereas Ok et al. (2016) prove that Belief Propagation (BP) is optimal, i.e. matches the performance of the MAP estimator, under the one coin model and when each worker is assigned at most 2 tasks. In our theoretical analysis, we focus on this regime. The key distinction of our work from the above body of work is that while existing work makes specific probabilistic assumptions on how workers label, we also allow for workers who differ from the probabilistic workers and adopt general strategies, whether probabilistic or not.

Detecting Unreliable/Adversarial workers. In addition, there are many works that aim to explicitly detect and/or remove unreliable workers based on the observed labels. One approach is to use "gold standard" tasks, i.e. tasks whose true label is already known, to identify low reliability workers (Snow et al., 2008; Downs et al., 2010; Le et al., 2010). However, getting access to task true labels can be hard 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) define scores customized to specific adversary strategies to identify and remove them. Similarly, Hovy et al. (2013) model the worker population as consisting of two types – one who always provide the correct label and spammers who provide uniformly random labels – and estimate the trustworthiness of each worker using the observed labels. Unlike these works, we allow the adversaries to adopt arbitrary strategies. Ipeirotis et al. (2010) proposed a way of 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 – higher the cost, lower the quality of the worker. Raykar and Yu (2012) propose an empirical Bayesian algorithm to eliminate workers whose labels are not correlated with the true label (called *spammers*), and estimate the consensus labels from the remaining workers. Both of these works rely on the Dawid-Skene model, whereas our algorithm does not rely on specific probabilistic assumptions for worker strategies. There have also been some attempts 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) consider malicious workers who can collude and provide arbitrary responses to degrade the performance of the aggregation algorithms and show 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 don't consider adversaries who can adapt their responses based on the labels submitted by the honest workers. Our analysis considers *arbitrary* worker-task assignment graphs and we allow the adversaries to choose their labels based on the observed honest worker responses.

Sybil attacks. Finally, our work is also broadly related to the rich literature of identifying *Sybil* identities in online social networks. Most of these schemes (Yu et al., 2006, 2008; Danezis and Mittal, 2009; Tran et al., 2011; Viswanath et al., 2012) make use of 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 since 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 $\mathcal{T} = \{t_1, t_2, \dots, t_m\}$ 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 $W = \{w_1, w_2, \dots, w_n\}$ 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 using a bipartite graph $\mathcal{B} = (W \cup \mathcal{T}, E)$ with the workers on one side, the tasks on the other side, and an edge $(w_i, t_j) \in E$ indicating that worker w_i was assigned task t_j. The graph $\mathcal{B}$ is termed the *worker-task assignment graph.* We suppose that the assignment $\mathcal{B}$ 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 her. 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 $1 \le i \le n$ and $1 \le j \le m$, where we set $w_i(t_j) = 0$ for any task t_j not assigned to worker w_i. We let $W_j \subseteq W$ denote 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 is comprised of 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 is the set of adversarial workers. The class memberships of the workers

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

are latent, so we do not know whether a worker is honest or not. Honest workers provide (noisy) labels according to some probabilistic model (such as the one-coin model introduced in Section 1). Adversarial workers are those whose labeling strategy does not conform to this probabilistic model and they can adopt arbitrary (deterministic or probabilistic) strategies. Examples include: (a) the *uniform strategy*, in which the worker arbitrarily provides the uniform response $+1$ or -1 to all the assigned tasks, irrespective of the true label; (b) the *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 the tasks are a priori known to have the label -1, then the worker chooses -1 as the uniform label and vice-versa; or (c) the *sophisticated strategy*, in which the worker adopts strategies specifically designed to cause the maximum "damage" (refer to section 4.2 for details). Note that all of these example strategies cannot be captured by the one-coin model.

We make the following remarks about our model. First, most existing work only focuses on the honest workers and our contribution is also considering adversarial workers. Second, our definition of the "adversary" is intentionally broader than common definitions to accommodate a wide range of labeling strategies. In fact, some of the example adversary strategies described above may be accommodated by extending the standard one-coin model. For instance, the uniform strategy can be accommodated by allowing the worker reliability parameter p_w to also depend on the true label of the task. The smart strategy can be accommodated by further allowing the parameters to depend on the population prevalence of the task labels. While such case-by-case extensions are feasible in theory, they result in custom algorithms with limited scope of application. In addition, 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* the adversarial workers. The adversaries identified by our algorithm may be filtered out or investigated further, depending on the application. More precisely, our objective is to solve the following problem:

Problem 1 *Given a set of workers $W = H \cup A$, tasks $\mathcal{T}$ and the 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 the adversaries. The algorithm relies on detecting workers whose labeling patterns are statistical *outliers* among the population of workers.

For the rest of the paper, we make the following technical assumptions. These assumptions are required *only* for proving theoretical guarantees, and the algorithm itself can be applied even if the assumptions don't hold. They also help in motivating the intuition behind our reputation algorithms. Suppose that the honest workers provide labels according to the one-coin model: for task t_j, honest worker h_i provides the correct label y_j with probability μ_i and the incorrect label $-y_j$ with probability $1 - \mu_i$, where μ_i is called the *reliability* of honest worker h_i. We assume that the honest workers are in majority, $|H| > |A|$, and their reliabilities are sampled from an underlying population with average reliability $\mu > \frac{1}{2}$. The assumption that $\mu > \frac{1}{2}$ is required for unique recovery of the true labels of the tasks, even with only honest workers (Karger et al., 2014). Further, in many crowdsourcing applications, the notion of ground truth is subjective and for such scenarios, a natural choice for defining the ground truth is what the majority of the population would agree on, had we asked the question to everyone in the crowd. Finally, the tasks are sampled from a population in which the *prevalence* of the positive label is $\gamma \in [0, 1]$, i.e. there is a fraction γ of tasks with true label $+1$. Note that our algorithms do *not* require knowledge of μ or γ.

3. Reputation Algorithms

We now describe the algorithm we propose to identify the adversarial workers, given the response matrix $\mathcal{L}$. We suppose that there is no side information on the identities of the workers (such as, say, a social network or worker-level demographic information), so the algorithm must solely rely on the response patterns given by the 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 the reputation of a worker, the algorithm relies on the number of *conflicts* the worker is involved in. Broadly speaking, a worker is involved in a conflict if her response to an assigned task is in disagreement with those of other workers. Note that tasks with a consensus opinion, having all $+1$ labels 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 the tasks that lack consensus, having both $+1$ and -1 labels. We term this subset of tasks as the *conflict set* $\mathcal{T}_{cs}$, and workers who respond to tasks in the conflict set are all involved in conflicts. A conflict typically signifies the presence of low reliability honest workers (who tend to make mistakes) or adversaries. In the ideal case when all honest workers are perfect, i.e., have reliabilities $\mu_i = 1$, 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 the 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. A simple counting of the number of conflicts then, may over-penalize honest workers who label a large number of tasks.

To overcome the issue of over-penalizing honest workers, we propose two penalty aggregation techniques, resulting in two variants of our algorithm: (a) *soft-penalty* and (b) *hard-penalty*. We describe these variants in detail next.

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 $1/d_j^+$ to all workers who provide the label $+1$ for t_j and $1/d_j^-$ to all workers who provide the label -1. 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 she is allocated a lower penalty than a worker who disagrees with the majority. Further, taking the average normalizes for the number of tasks labeled by any worker. The algorithm relies on the following intuition in allocating the penalties: assuming the average reliability of the honest workers $\mu > \frac{1}{2}$, we expect that on an average, the honest workers provide the correct response to the assigned tasks. Further, 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 of the 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 (refer to Theorems 2, 3, 5 and 9). Our results demonstrate asymptotic consistency of the soft-penalty algorithm in identifying the adversaries, under standard assumptions on the structure of the worker-task assignment graph.

Even though the soft penalty algorithm is successful in identifying adversarial workers adopting certain types of strategies, its performance depends on the complexity of the strategies employed by the adversaries. If the adversarial workers are non-colluding and adopt non-deliberate strategies, then the soft penalty algorithm can identify them from the observed responses. The algorithm, however, is subject to manipulation 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 and therefore is susceptible to attack by determined adversaries who can cleverly decide their responses based on the honest worker labels and the structure of the worker-task assignment graph, to cause maximum "damage". As a concrete example, suppose that the subgraph of $\mathcal{B}$ between honest workers and tasks is r right-regular, i.e. 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 (i.e., $\mu_i = 1$ for each honest worker h_i). Now, suppose there are $k > r$ adversaries and each adversary provides the incorrect response to *all* the tasks. Then, every task has r correct responses, all provided by honest workers, and k incorrect responses, all provided by adversaries, resulting in 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 penalty because the penalty received from each task is the same). Because $k > r$, the adversaries receive lower penalties than the honest workers. As a result, filtering out k workers with the highest penalties will always filter out the honest workers. Furthermore, natural aggregation algorithms (simple majority or weighted majority with penalties as weights) result in incorrect labels for all the tasks.

In fact, for such worst-case adversaries, we show that (see Theorem 12) for any collection of honest worker responses, there exists 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 (as long as the aggregation algorithm satisfies a few natural properties). To account for such adversarial behavior, we introduce the *hard-penalty* algorithm next.

3.2 Hard Penalty

To address the case of these 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 the other workers on *all* the other tasks. In particular, instead of distributing the penalty evenly across all the workers that respond to a given task, the algorithm chooses two 'representative' workers to penalize per conflict task: one representative worker among those who provide the label $+1$ and another among those who provide the label -1. The choice of the representative workers is done in a load-balanced manner to "spread" the penalty across all the workers, so that we don't over-penalize workers who provide labels for a large number of tasks. The net penalty of each worker is the sum of the accrued penalties across all the (conflict) tasks assigned to the 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 the representative workers in load-balanced fashion, we use the concept of *optimal semi-matchings* (Harvey et al., 2003) on 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, i.e. the number of edges in M that are incident to u and let $cost_M(u)$ be defined as

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

An *optimal semi-matching* then, is one which minimizes $\sum_{u \in V_1} cost_M(u)$. Intuitively, an optimal semi-matching *fairly* matches the V_2-vertices across the V_1-vertices so 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 the 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 serial order. An optimal semi-matching can be thought of as an assignment of the tasks to the machines such that the *flow-time*, i.e. the average completion time of a task, is minimized. Refer to (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^+ if the worker labeled the task $+1$ and to t_j^- if the worker labeled the task -1. The optimal semi-matching, by definition, yields two representative workers for each task t_j – one connected to t_j^+ and the other connected to t_j^-. As for the case of soft-penalty, we only consider conflict tasks when creating this augmented bipartite graph. The worker degrees in the optimal semi-matching then constitute their net penalties. The hard-penalty algorithm is described in detail in Algorithm 2.

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, $pen(w_i) = deg_{\mathcal{M}}(w_i)$

3.3 Connection between soft-penalty and hard-penalty algorithms

While the hard and soft-penalty algorithms appear different on surface, the soft-penalty algorithm can be interpreted as the random, normalized variant of the hard-penalty algorithm. 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 the workers are assigned the same number of tasks, the expected penalty assigned by the random hard-penalty algorithm 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 a 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 the adversaries from the 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 theorems are given in the appendix.

4.1 Soft-penalty algorithm: common adversary strategies

We analyze the performance of the soft-penalty algorithm under two settings: (a) the classical setting when there are no adversaries ($A = \emptyset$) and (b) when the adversaries adopt the Uniform strategy.

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

We consider the Uniform strategy because of its ubiquity and simplicity. It is commonly observed in practice (Vuurens et al., 2011) but 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 its 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, leading a "smart" worker to label all the assigned images as benign without carefully considering each image. For this reason, the Uniform strategy, in fact, comprises a spectrum of strategies of varying degrees of "smartness" with higher values of γ indicating smarter strategies.

As the performance of the algorithm depends on the specific crowdsourced classification instance (worker-task assignment graph, task true labels, reliabilities of honest workers), we carry out 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 proportion $q \in (0, 1]$ of honest workers, the number of tasks m, the number of workers n, the worker degree $l > 1$, the task degree $r > 1$ are fixed. Let $\gamma \in [0, 1]$ denote the prevalence or the proportion of tasks with true labels $+1$ and $F(\cdot)$ denote the cumulative distribution function (CDF) of the honest worker reliabilities with $\mu \in [0, 1]$ denoting the mean. Sample a crowdsourced classification instance as follows:

1. *Worker-task assignment graph:* assign the 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 the worker half-edges and map them to the task half-edges.

2. *Task true 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 the remaining probability $1 - q$.

4. *Honest worker reliabilities and responses:* if w_i is honest, sample its reliability $\boldsymbol{M}_i = \mu_i$ from 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 the random graph literature (Bollobás, 2001). It may result in a graph with multi-edges (where two nodes are connected by more than one edge), but the number of double-edges converges in distribution to the Poisson distribution with mean $(l-1)(r-1)/2$ (Bollobás, 2001, Page 59, Exercise 2.12). Therefore, the proportion of the nodes with multi-edges is $\approx lr/n$, which goes to zero as $n \to \infty$ as long as $l = o(n)$ and r is constant.

The model for the task true 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 the 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 task true 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 the 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 the 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 the task will be $1/r$, 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 for details).

4.1.1 Analysis of expected penalties

We first analyze the expected penalties received by the honest and adversarial workers under the generative model above, and identify the conditions on the 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 generative model above.

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

Theorem 2 (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 2 shows that the expected reputation scores are consistent with honest worker reliabilities: as the honest worker reliability decreases, the expected penalty increases or in other words, 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 when $A \neq \emptyset$ 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; because of symmetry, these expectations do not depend on worker indices. Then, we have the following result:

Theorem 3 (Penalties under Uniform strategy) *When $q < 1$ and the adversaries adopt the Uniform strategy, we have*

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

where $h(\mu, q)$ is a strictly increasing function in q for a fixed μ, 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 on the parameters q, μ, and γ under which the soft-penalty algorithm is successful in assigning lower penalties to honest workers than the adversaries.

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

Taking the above observation further, 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, must exceed $\frac{1}{2q}$ (for a fixed q). To obtain an intuitive understanding of this condition (the precise proof is given in the appendix), note the following. Consider a task with true label $+1$. Then, in expectation, there $rq\mu$ 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 $r\cdot(q\mu+(1-q))\geq rq\cdot(1-\mu)$, i.e., $\mu\geq 1-1/(2q)$. 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 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 hence, 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 $\mu>\frac{1}{2q}$, i.e., $q\mu>\frac{1}{2}$.

Now assuming that the first condition is met, i.e., $q\mu>1/2$, we focus on the second condition: $\frac{\mu}{1-\mu}\cdot h(\mu,q)>\frac{\gamma}{1-\gamma}$. When $\gamma=1$, the condition is not met (unless $\mu=1$) and, therefore, honest workers receive higher expected penalties than adversaries. The reason is that if all the tasks have a true label $+1$, then in expectation, there is a fraction $q\mu+1-q>1/2$ (because $q\mu>1/2$) 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 the adversaries in fact have higher (perfect) reliabilities in this special case. In a similar fashion, when $\mu=1$, i.e., the honest workers are perfectly reliable, the condition is always met because the honest workers are in majority at each task (in expectation); specifically, when the true label is $+1$, all the responses are $+1$ and when the true label is -1, all the 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 4 (Penalties under smarter adversary strategies) *Suppose $\mu>\frac{1}{2}$ is fixed and $q\mu>\frac{1}{2}$. Then,*

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

where $h_\mu(q):=\frac{\mu}{1-\mu}\cdot h(\mu,q)$ and $h_\mu^{-1}(\cdot)$ is the inverse of $h_\mu(\cdot)$, which is well-defined because h_μ is a strictly increasing function.

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

4.1.2 Asymptotic identification of adversaries and honest workers

Assuming that the expected penalties of the 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.

In order 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

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

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 when there are no adversaries, i.e., $q = 1$, so that $\boldsymbol{I}(w_i) = \textit{honest}$ for all workers w_i. In this case, Theorem 2 above shows that workers with higher reliabilities receive lower expected penalties. Building on this, we now show that the threshold-classifier correctly classifies high-reliability workers as honest with high probability, as $n \to \infty$. More precisely:

Theorem 5 (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 3. Then, under the generative model, we have*

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

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

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

Theorem 5 provides an upper bound for the *error rate* or the *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, $l^2r^2/(n-1)$, 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 $n \to \infty$. 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 go to infinity, it must only grow logarithmically in the total number of workers n. Finally, if the population reliability μ is known, we can back out the value of threshold θ for a given reliability threshold $\hat{\mu}(\theta)$.

The proof of Theorem 5 relies on establishing that the worker penalties concentrate around their respective expectations, for which we need the worker-task assignment graph $\mathcal{B}$ to be locally tree-like:

Definition 6 (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 at most D from w_i, is a tree.*

For our purposes, it suffices to have $\mathcal{B}_{w_i,2}$ to be a tree. Note that the subgraph $\mathcal{B}_{w_i,2}$ consists of the worker node w_i, the tasks labeled by w_i, i.e. the set $\mathcal{T}_i$, and the 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 7 (Random construction ensures local-tree structure) *If $\mathcal{B}$ is a random (l, r)-regular graph constructed according to the configuration model, then for a randomly chosen worker $\boldsymbol{w}$,*

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

The proof of the above lemma is in the appendix and follows along the lines of the arguments in Karger et al. (2014).

Based on the result of Lemma 7, the proof of Theorem 5 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 a probability of error 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 8 (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 the 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 the worker node w_i ensures that the l random variables are mutually independent (which is required for the Hoeffding's inequality).

Next, we consider the case when there is a fraction $1-q$ workers who are adversaries and adopt the $\mathsf{Uniform}$ strategy. Theorem 3 above provided the necessary and sufficient conditions for the honest worker to receive a lower expected penalty than the adversary, i.e., for $p_h < p_a$. Under these conditions, we have the following result:

Theorem 9 (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 have:*

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

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

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 have

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

The precise expression for $\hat{\mu}(q,\theta)$ is involved and is given in the appendix. Theorem 9 provides the misclassification rate of our algorithm when the population parameters q, μ, and γ satisfy the conditions of Theorem 3, ensuring that the honest workers receive a lower expected penalty than the adversaries. Following the arguments from the discussion under Theorem 5 above, it can be seen that when $l = \log n$ and r is fixed, the fraction of adversaries that are misclassified is $O(1/n^{2\varepsilon^2})$. On the other hand, the fraction of honest workers that are misclassified scales as $O(1/n^{2\varepsilon^2}) + F(\hat{\mu}(q,\theta))$. The first term goes to zero as $n \to \infty$. The second term denotes the probability that the 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 goes to zero as $n \to \infty$.

We note that the dependence of the honest worker misclassification rate on $F(\hat{\mu}(q,\theta))$ 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 17 in the appendix that

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

It follows from Theorem 3 that $P > 1/2$ and $Q < 1/2$ is necessary to ensure that $p_h < p_a$. 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 that $p_1 < p_a < p_0$. It now follows that, when $n \to \infty$, the penalties of the honest workers with reliabilities 0 concentrate around p_0, and consequently, are classified as adversarial whenever the threshold $\theta < p_a$, resulting in a misclassification error of $1-\mu$. Note, however, that the honest workers classified as adversaries indeed have low reliabilities.

Similar to the proof of Theorem 5 above, the proof of Theorem 9 proceeds in two steps. The first term in the error bound of Theorem 9 comes from Lemma 7 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 10 (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:*

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

and

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

The proof of the above result is similar to that of Lemma 8. For the case of adversarial workers, we use a Hoeffding's argument to establish the concentration. For the case of honest workers, the first term follows directly from Lemma 8 when w_i has a reliability $\mu_i > \hat{\mu}$ and the second term is the probability that the reliability $\mu_i \leq \hat{\mu}$.

The above results show that the soft-penalty algorithm successfully identifies low reliability honest workers and adversaries adopting the Uniform strategy asymptotically, with high probability. Note that all of the 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 (Refer to Appendix A.3 for more details).

4.2 Hard-penalty algorithm: sophisticated adversary strategies

In the preceding analysis, we focused on common adversary strategies in which the adversaries were not intentionally malicious. However, existing work provides 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 on the Web by an attacker (Wang et al., 2012) to create fake accounts and manipulate the ratings/reviews with the purpose of altering the aggregate ratings or rankings received by the 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), malicious crowd behavior in online surveys (Gadiraju et al., 2015). See the recent work by Wang et al. (2014) for more examples. Motivated by these examples, we study settings with *sophisticated adversaries*, defined as follows:

Definition 11 (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, colluding, and possess knowledge of the labels provided by the honest workers; therefore, they can adopt arbitrary response strategies.*

Our definition allows the sophisticated adversaries to not just be malicious but capable of executing the most complex strategies. In practice, the adversary may adopt feasible strategies with varying complexities, depending on the application context and their objectives. But focusing on the most sophisticated adversary makes our analysis broadly applicable, independent of the application context. Further, we don't 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.

For our analysis, we measure the performance of the hard-penalty algorithm in terms of *accuracy*, the number of tasks that receive correct labels, as opposed to their ability to identify the adversaries. This is done to gain analytical tractability.

Before we analyze the performance of our algorithm, we prove a lower bound on the number of tasks that will receive incorrect labels, irrespective of the label aggregation algorithm employed to aggregate the worker responses. The lower bound provides a way to assess the optimality of the hard-penalty algorithm.

4.2.1 Lower bound on the 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 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 the 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 task true 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 i.e., each honest worker h_i has $\mu_i = 1$ and always provides the correct response. Focusing on completely reliable honest workers allows us to isolate the impact of adversaries because any misidentification is caused due to 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. Since 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 have the following result, which establishes a lower bound on the quality of any decision rule:

Theorem 12 (Lower bound on number of affected tasks) *Suppose that* $|A| = k$ *and* $\mu_i = 1$ *for each honest worker* $h_i \in H$. *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*

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

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

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 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* $\boldsymbol{R} \in \mathcal{C}$.

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 2 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, the 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 since 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 hence 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 12 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 prior works; see Dalvi et al. (2013)).

The bound L itself depends on the structure of $\mathcal{B}_H$ and therefore can be hard to interpret in general. It, however, becomes interpretable for an (r, γ, α)-bipartite expander, defined next.

Definition 13 *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, i.e. 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 $|\mathrm{PreIm}(\mathcal{T}')| \geq \alpha\, |\mathcal{T}'|$, where $\mathrm{PreIm}(\mathcal{T}')$ 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 12 when $\mathcal{B}_H$ is (r, γ, α)-bipartite expander.

Corollary 14 (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 $\lceil \frac{k}{\alpha} \rceil + 1 < \gamma \cdot |\mathcal{T}|$. Further, 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.4. 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 the 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 14 holds for all r-right-regular graphs, even if they are not expanders.

4.2.2 Accuracy of the hard-penalty algorithm

We now analyze the accuracy of the hard-penalty algorithm when there are k sophisticated adversaries. We focus on the hard-penalty algorithm because it can be seen that the soft-penalty algorithm is vulnerable to sophisticated adversaries (see the example at the end of section 3.1). For our analysis, we focus on the *penalty-based aggregation* algorithm (see Algorithm 3), which is a natural extension of the hard-penalty algorithm to also perform label aggregation:

Algorithm 3 penalty-based aggregation

1: **Input:** W, $\mathcal{T}$ and $\mathcal{L}$
2: Perform steps 2,3 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 output label for task t_j and $\leftarrow \{-1, +1\}$ means $\hat{y}_j$ is drawn uniformly at random from $\{-1, +1\}$)

Theorem 15 (Hard-penalty algorithm with sophisticated adversaries) *Suppose that $|A| = k$ and $\mu_i = 1$ for each honest worker, i.e an honest worker always provides the correct label. Further, let $d_1 \geq d_2 \geq \cdots \geq d_{|H|}$ denote the degrees of the honest workers in the optimal semi-matching on $\mathcal{B}_H$. For any true labeling $\boldsymbol{y}$ of the tasks and under the penalty-based label aggregation algorithm (with the convention that $d_i = 0$ for $i > |H|$):*

1. *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. *No adversary strategy can affect more than U tasks where*
   (a) *$U = \sum_{i=1}^{k} d_i$, when all but (at most) one adversary are required to provide incorrect responses*
   (b) *$U = \sum_{i=1}^{2k} d_i$, 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.6) and therefore, the bounds in the theorem above are uniquely defined given $\mathcal{B}_H$. Also, the assumption that $\mu_i = 1$ is required for analytical tractability; proving theoretical guarantees in crowdsourced settings (even without adversaries) for general graph structures is notoriously hard (Dalvi et al., 2013). The result of Theorem 15 provides both a lower and upper bound for the number of tasks that can be affected by k adversaries under the penalty-based aggregation algorithm. Our characterization is reasonably tight when all but one adversary are required to 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 k large enough. However, our characterization is loose in the general case when adversaries can provide arbitrary responses; here the gap is $\sum_{i=k}^{2k} d_i$ which we attribute to our proof technique and conjecture that the upper bound of $\sum_{i=1}^{k} d_i$ also applies to the more general case.

Optimality of Penalty-based Aggregation. We now compare the upper bound U in Theorem 15 to the lower bound L in Theorem 12. We show that (see Appendix A.7) when the degrees $d_1, d_2 \ldots, d_{|H|}$ are all distinct, $L \geq \frac{1}{2}\sum_{i=1}^{k-1} d_i$, which combined with theorem 12 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. We also have from Theorem 15 that under the penalty-based aggregation algorithm, k adversaries can affect at most $U = \sum_{i=1}^{2k} d_i \leq 3(\sum_{i=1}^{k-1} d_i)$ (as long as $k \geq 2$). Therefore, our algorithm achieves constant factor optimality in recovering the true labels of the tasks, *irrespective* of the structure of the honest worker-task assignment graph *and* the adversary 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. Using standard crowdsourcing data sets (detailed below), it shows that filtering out the adversaries identified by our methods allows existing label aggregation algorithms to infer the task true labels more accurately. Such improvements in accuracies from *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 utilized by fitting a model different from that of the 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-reliable 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 task true label) and workers adopting the Uniform strategy. It demonstrates that both soft-penalty 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, complementing the results of Theorem 2 and Theorem 3, which establish a similar result for (l, r)-regular worker-task assignment graphs.

For the purposes of our studies, we focused on the following four label aggregation algorithms: (a) simple majority algorithm MV (b) the EM algorithm (Raykar and Yu, 2012) (c) the KOS algorithm (Karger et al., 2014) and (d) KOS(NORM), a normalized variant of KOS in which the messages are scaled by the corresponding worker and task degrees to account for non-uniform node degrees in the assignment graph. We implemented the two 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 pre-specified number of workers were removed.

Finally, as mentioned in Section 4.1, we implemented both the variants of the soft-penalty algorithm: one in which the non-conflict tasks are dropped for the purposes of assigning worker penalties and the other 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.

5.1 Accuracy improvements on real-world crowdsourcing data sets

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 (AMT); see Tang and Lease (2011). These data sets were collected as part of the TREC 2011 crowdsourcing track.

- `rte` and `temp` – consisting of annotations by AMT 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` consists of binary judgments 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).

As inferring the reliabilities of workers who labeled very few tasks [7] is difficult, we pre-processed the datasets to remove all workers who labeled less than three tasks. Table 1 provides the summary statistics of the datasets after our pre-processing.

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

Table 2 reports the accuracies of inferring task true labels of various label aggregation algorithms. For each benchmark label aggregation algorithm (MV, EM, KOS, and KOS(NORM)), the column BASE reports the accuracy of the algorithm in isolation. The columns SOFT and HARD report the best accuracy of the algorithms on removing $k = 1, 2, \ldots, 10$ workers using the soft-penalty and hard-penalty algorithms, respectively; the numbers in the parentheses are the value of k for which we observed the best performance.

Dataset	MV			EM			KOS			KOS(NORM)		
	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD
rte	91.9	92.1(7)	**92.5(3)**	93.0	93.0	**93.3(5)**	49.7	89.0(10)	**91.6(10)*** **	91.3	92.6(5)	**93.1(6)** **
temp	93.9	93.9	**94.4(5)**	94.1	94.1	94.1	56.9	69.3(4)	**93.7(3)*** **	93.9	94.4(7)	**94.4(1)**
stage2	74.3	75.4(1)	**80.5(2)*** **	70.2	76.8(4)	**81.2(6)*** **	74.5	74.7(7)	**75.1(2)**	75.5	75.5	**78.2(2)***
task2	64.2	64.2	**67.8(10)*** **	67.0	67.1(6)	**68.6(9)*** **	57.4	57.4	**65.6(10)*** **	58.3	58.9(8)	**67.7(9)*** **
tweets	69.6	69.8(4)	**73.3(1)*** **	71.2	71.2	**71.7(1)**	65.8	66.0(4)	**70.5(1)*** **	68.7	68.7	**71.0(2)*** **
aggregate	78.8	79.1	**81.7**	79.1	80.4	**81.8**	60.9	71.3	**79.3**	77.5	78.0	**80.9**

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 the 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 last row reports the average accuracy across the datasets. The p-values, according to a 2-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 is boosting 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(NORM) is 4.4% while for KOS it is 30.2%, when using the hard-penalty algorithm.

7. The datasets `stage2`, `task2` and `tweets` contain several workers who provided responses for only a single task.

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 are able to boost the performance of MV and KOS algorithms to the level of the popular EM algorithm. The MV algorithm is simple to implement, and the KOS algorithm is shown to have strong theoretical guarantees when the underlying assignment graph is random (l, r)-regular and to be robust to different initializations (Karger et al., 2014). Our results suggest that implementing MV and KOS algorithms in conjunction with our reputation algorithms can allow us to obtain their respective simplicity and theoretical guarantees along with strong practical performance, comparable to that of the EM algorithm. Finally, because discarding labels of certain workers is improving predictive accuracy, our results suggest that standard probabilistic models are insufficient to capture the labeling patterns of workers in real-world data sets.

In addition to the four label aggregation algorithms described above, we carried out the above analysis for two recently proposed benchmarks – spectral EM by Zhang et al. (2014) and the regularized minimax conditional entropy approach by (Zhou et al., 2015) – that have been shown to perform well on real data sets. The results are qualitatively similar, and we refer an interested reader to Appendix B for details.

To gain insights into the types of workers identified by our algorithms, we did 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% were the same label. For instance, our algorithms detected 6 such workers in the `temp` dataset, 5 in the `task2` dataset, and 1 in the `tweets` dataset. For the `stage2` data set, we detected 2 workers who gave all $+1$ labels and one worker who gave all but a single response as -1; it is worthwhile to note that the empirically computed prevalence γ of $+1$ tasks in the `stage2` data set was 0.83, potentially suggesting that the 2 workers who gave all $+1$ labels were adopting "smart" strategies.

2. Workers who provide labels independent of task true labels. For instance, we detected 4 such workers in the `rte` dataset, 7 workers in the `temp` dataset, 7 in the `task2` data set, 3 in `stage2`, and 1 in the `tweets` data set, 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) show 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: accurate on tasks with one type of true label, say, $+1$, but not on the 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 4 workers in the `tweets` data set and 2 workers in `stage2` data set that have skewed reliabilities. In the `tweets` data set, 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 tasks. A similar explanation applies to the `stage2` data set, 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 by our algorithms were on average *lower* than those of the unfiltered workers. This suggests that the labels our algorithms discard are from low-reliable workers, who provide little to no information about the task true labels – offering an explanation for the accuracy improvements we obtain.

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 of 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*: the fraction of filtered-out workers who are adversarial and have low empirical reliabilities.

Setup of the study. We considered a total of $n = 100$ workers. The probability q that a worker is honest was set to 0.7, so that 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 used the Python networkx library (Hagberg et al., 2008) to generate the worker-task assignment graph.[8] 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; (d) for each honest worker w, sample her reliability μ_w u.a.r from the interval $[0.8, 1.0)$, and set her response

8. 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 the low and high degree workers, respectively, are more likely to be adversaries. Refer to the main text for more details.

to each assigned task t_j to be the task true label y_j with probability μ_w and $-y_j$ with probability $1-\mu_w$; (f) 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 2 because it is captured by the one-coin model, and the second strategy reflects the setting of Theorem 3.

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 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 her responses that were correct. The table reports the precisions 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 precisions 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-reliable honest workers when the worker and task degrees have a power-law distribution. This finding complements the results of Theorem 5 and Theorem 9, which establish a similar result for 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 speaking, 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 the 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 paper has studied 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. The honest workers may be reliable or unreliable, and provide labels according to a well-defined probabilistic model. The adversaries adopt strategies different from those of the 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 – the soft-penalty and the hard-penalty algorithms – that analyze the worker label patterns and assign a reputation score to each worker to indicate the degree to which the worker label 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 out adversaries from honest workers when the adversaries adopt deterministic strategies. When the 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 significantly enhance the accuracy of existing label aggregation algorithms on 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 a interesting direction. Because our algorithm allows identification of adversarial workers, it may be combined with adaptive techniques to recruit more workers (Ramesh et al., 2012) and identifying the right workers (Li et al., 2014). Finally, applying our outlier detection technique to ensemble learning approaches, where 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).

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. International World Wide Web Conferences Steering Committee, 2013.

George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. NDSS, 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. 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:1310.5764*, 2013.

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*, 2015.

David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. Neural Information Processing Systems, 2011.

David R Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, pages 81–92. 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. Reliable crowdsourcing under the generalized dawid-skene model. *arXiv preprint arXiv:1602.03481*, 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 crowd-sourced 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 25*, 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. International World Wide Web Conferences Steering Committee, 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 14–14. 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. *The 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. *The Journal of Machine Learning Research*, 99:1297–1322, 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. *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. 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 *INFOCOM, 2011 Proceedings IEEE*, pages 3218–3226. IEEE, 2011.

Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 363–374. ACM, 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, CA*, 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. *Advances in Neural Information Processing Systems*, 23:2424–2432, 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. *Advances in Neural Information Processing Systems*, 22(2035-2043):7–13, 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(4):267–278, 2006.

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

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 $\text{Bin}(n,p)$ 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 16 *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 $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = +1 \sim \text{Bin}(r, P)$ and $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = -1 \sim \text{Bin}(r, Q)$.

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 17 *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 16 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 16 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 2

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:

$$\begin{aligned}
&\mathbb{E}[\mathbf{PEN}_i \,\Big|\, \boldsymbol{M}_i = \mu_i] = \frac{1}{l}\sum_{j \in \mathcal{T}_i} \mathbb{E}[\boldsymbol{S}_{ij} \,\Big|\, \boldsymbol{M}_i = \mu_i] \\
&= \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i) \cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i) \cdot g(1-Q)\Big) \\
&\text{(using lemma 17)} \\
&= \gamma\Big(\mu_i \cdot g(1-\mu) + (1-\mu_i) \cdot g(\mu)\Big) + (1-\gamma)\Big(\mu_i \cdot g(1-\mu) + (1-\mu_i) \cdot g(\mu)\Big) \\
&= g(\mu) - \mu_i \cdot (g(\mu) - g(1-\mu))
\end{aligned}$$

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 3

For an honest worker w_i, we have that

$$
\begin{aligned}
p_h &:= \mathbb{E}_{ih}[\mathbf{PEN}_i] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\mathbb{E}_{ih}[\boldsymbol{S}_{ij}] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\int_0^1 \mathbb{E}_{ih}[\boldsymbol{S}_{ij}\,\Big|\,\boldsymbol{M}_i=\mu_i]f(\mu_i)d\mu_i \quad \text{(law of total expectation)} \\
&= \int_0^1 \Big(\gamma\cdot(\mu_i\cdot g(1-P)+(1-\mu_i)\cdot g(P))+(1-\gamma)\cdot(\mu_i\cdot g(Q)+(1-\mu_i)\cdot g(1-Q))\Big)f(\mu_i)d\mu_i \\
&\text{(using lemma 17)} \\
&= \gamma\Big(\mu\cdot g(1-P)+(1-\mu)\cdot g(P)\Big)+(1-\gamma)\Big(\mu\cdot g(Q)+(1-\mu)\cdot g(1-Q)\Big) \\
&\left(\text{since } \int_0^1 \mu_i f(\mu_i)d\mu_i=\mu\right)
\end{aligned}
$$

Similarly, when worker w_i is adversarial,

$$
\begin{aligned}
p_a &:= \mathbb{E}_{ia}[\mathbf{PEN}_i] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\gamma\cdot g(1-P)+(1-\gamma)\cdot g(1-Q) \quad \text{(using lemma 17)} \\
&= \gamma\cdot g(1-P)+(1-\gamma)\cdot g(1-Q)
\end{aligned}
$$

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

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

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:

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

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 4

Since $q\mu > \frac{1}{2}$, we have that $P > \frac{1}{2}$ and $Q < \frac{1}{2}$. From the proof of theorem 3 above, we have

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

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 7

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, atleast 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 8

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

$$\Pr[\boldsymbol{W}_{ij} = v_1; \boldsymbol{W}_{ij'} = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i] =$$

$$\sum_{(x_1,x_2)\in\{-1,+1\}} \Pr[\boldsymbol{W}_{ij} = v_1; \boldsymbol{W}_{ij'} = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2]\Pr[\boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2] =$$

(since the task true labels are independent of $\boldsymbol{M}_i$)

$$\sum_{(x_1,x_2)} \Pr[\boldsymbol{W}_{ij} = v_1 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = x_1; \boldsymbol{W}_{ij'} = v_2]\Pr[\boldsymbol{W}_{ij'} = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = x_2]\Pr[\boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2]$$

$$= \left(\Pr[\boldsymbol{W}_{ij} = v_1 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = +1]\Pr[\boldsymbol{Y}_j = +1] + \Pr[\boldsymbol{W}_{ij} = v_1 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = -1]\Pr[\boldsymbol{Y}_j = -1]\right)\times$$

$$\left(\Pr[\boldsymbol{W}_{ij'} = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = +1]\Pr[\boldsymbol{Y}_{j'} = +1] + \Pr[\boldsymbol{W}_{ij'} = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = -1]\Pr[\boldsymbol{Y}_{j'} = -1]\right)$$

$$= \Pr[\boldsymbol{W}_{ij} = v_1 \,\Big|\, \boldsymbol{M}_i = \mu_i] \cdot \Pr[\boldsymbol{W}_{ij'} = v_2 \,\Big|\, \boldsymbol{M}_i = \mu_i]$$

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 5

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 2 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

$$\begin{aligned}
&\Pr(\mathbf{PEN}_{\boldsymbol{w}} > \theta \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)) \\
&= \Pr(\mathbf{PEN}_{\boldsymbol{w}} > p_{\hat{\mu}(\theta)} + \varepsilon \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)) \\
&= \int_{\hat{\mu}(\theta)}^{1} \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} > p_{\hat{\mu}(\theta)} + \varepsilon \,\Big|\, \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&\leq \int_{\hat{\mu}(\theta)}^{1} \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} > \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}] + \varepsilon \,\Big|\, \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&(\text{since } \tilde{\mu} > \hat{\mu}(\theta) \implies \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}] < p_{\hat{\mu}(\theta)}) \\
&\leq \int_{\hat{\mu}(\theta)}^{1} \exp\left(\frac{-2l\varepsilon^2}{(1 - 1/r)^2}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&(\text{from lemma 8 above}) \\
&\leq \exp\left(\frac{-2l\varepsilon^2}{(1 - 1/r)^2}\right)
\end{aligned}$$

Finally, combining all the claims above, we have

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

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 9

We follow a similar line of reasoning to that of theorem 5. 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 17. 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

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

The claim then follows from the result of lemma 10 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 10 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 5 above.

A.2.5 Proof of Lemma 10

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):

$$\begin{aligned}
&\Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \Big| w_i \text{ is honest }) \\
&= \int_0^1 \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \Big| w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&= \int_0^{\hat{\mu}} \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \Big| w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&+ \int_{\hat{\mu}}^1 \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \Big| w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&\leq \int_0^{\hat{\mu}} f(\mu_i) d\mu_i \\
&+ \int_{\hat{\mu}}^1 \Pr(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \epsilon \Big| w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&(\text{ since } \mu_i \geq \hat{\mu} \implies \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] \leq p_{\hat{\mu}}) \\
&\leq F(\hat{\mu}) + \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) \\
&(\text{ from lemma 8})
\end{aligned}$$

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 8 and 10 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 Theorem 12

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

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

For a given decision rule $R \in \mathcal{C}$, we construct an adversary strategy σ^* under which atleast L tasks are affected for some true labeling of the tasks. 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)

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

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 and not use the remaining. 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 $\boldsymbol{y}$.

Now consider the scenario in which the true labels of all tasks in $\mathcal{T}^*$ are flipped, 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:

$$\tilde{h}(t) = \begin{cases} h(t) & \forall\, t \notin \mathcal{T}^* \\ -h(t) & \forall\, t \in \mathcal{T}^* \end{cases} \qquad \forall\, h \in H \tag{1}$$

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

$$\tilde{a}(t) = \begin{cases} a(t) & \forall\, t \notin \mathcal{T}^* \\ -a(t) & \forall\, t \in \mathcal{T}^* \end{cases} \qquad \forall\, a \in A \tag{2}$$

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. For that, 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 (1) 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 (1). Therefore, we have that $r(h) = \tilde{r}(h)$ for any $h \notin H^*$. Next, consider an honest worker $h_i \in H^*$ for some i. It can be argued that $r(h_i) = \tilde{r}(a_i)$. To see this, for any task $t \notin \mathcal{T}^*$, we have by (2) that $\tilde{a_i}(t) = a_i(t) = h_i(t)$, where the second equality follows from our definition of the adversary strategy. Similarly, for any $t \in \mathcal{T}^*$, we have $\tilde{a_i}(t) = -a_i(t)$ by (2) and $a_i(t) = -h_i(t)$ (by the adversary strategy). Hence, we must have $\tilde{a_i}(t) = h_i(t)$ for any $t \in \mathcal{T}^*$. Thus, we have shown that the rows $\tilde{r}(a_i) = r(h_i)$ for any i. Thus, $\mathcal{L}(\tilde{\boldsymbol{y}})$ 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 $R\left(\mathcal{L}(\tilde{\boldsymbol{y}})\right) = R\left(\mathcal{L}(\boldsymbol{y})\right)$. 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 atleast L, and since we take a maximum over all possible ground-truth labelings, the result of the theorem follows.

Remark. 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.

A.4.1 Proof of Corollary 14

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 atmost 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 [EXP].

A.5 Proof of Theorem 15

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

Definition 18 *A bipartite graph $G = (V_1, V_2, E)$ is termed* **degenerate** *if the following condition is satisfied:*

$$|V_1| > |V_2|$$

Definition 19 *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 20 *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 21 *Let $G = (V_1, V_2, E)$ be any bipartite graph and suppose that M is* any *semi-matching on G. Further, let $J = (V_1, V_2', E')$ be a subgraph of G. Starting with $M' \subseteq M$, we can use algorithm $\mathcal{A}_{SM2}$ in [OPT] to obtain an optimal semi-matching N for the subgraph*

J. Let the nodes in V_1 be indexed such that $deg_M(1) \geq deg_M(2) \geq \dots deg_M(|V_1|)$ and indexed again such that $deg_N(1) \geq deg_N(2) \geq \dots deg_N(|V_1|)$. Then for any $1 \leq s \leq |V_1|$, we have $\sum_{i=1}^{s} deg_N(i) \leq \sum_{i=1}^{s} deg_M(i)$, i.e. the sum of the top s-degrees can only decrease as we go from M to 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 $P = (v_1^{(1)}, v_2^{(1)}, v_1^{(2)}, \dots v_1^{(d)})$ be a cost-reducing path (see section 2.1 in [OPT]). This means that $deg(v_1^{(1)}) > deg(v_1^{(d)}) + 1$. 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 $d_1^{\text{bef}} \geq d_2^{\text{bef}} \geq \dots d_{|V_1|}^{\text{bef}}$ and $d_1^{\text{aft}} \geq d_2^{\text{aft}} \geq \dots d_{|V_1|}^{\text{aft}}$ be the degree-sequence 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

$$\sum_{i=1}^{s} deg_N(i) \leq \sum_{i=1}^{s} deg_{M'}(i) \tag{3}$$

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.

$$\sum_{i=1}^{s} deg_{M'}(i) \leq \sum_{i=1}^{s} deg_M(i) \tag{4}$$

Combining equations (3) and (4), the result 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. Recall the conflict graph $\mathcal{B}^{cs}$ constructed in Algorithm 2, where we similarly created two copies for each task. Now, since honest workers always provide the correct response, all honest workers have edges *only* to the set of task copies in E, in $\mathcal{B}^{cs}$. However, adversaries can have edges to task copies in both E and F. In addition, it is easy to see that the subgraph of $\mathcal{B}^{cs}$ restricted to honest workers H on the left and task copies E on the right, has exactly the same structure as $\mathcal{B}_H$. As a result, the optimal semi-matching M_E over the sub-graph is the same as the optimal semi-matching on the bipartite graph $\mathcal{B}_H$, which we denote by M_H. Thus, the degrees of the honest workers in M_E are by hypothesis of the theorem $d_1, d_2, \dots, d_{|H|}$. Without loss of generality, suppose that honest workers are indexed such that $d_1 \geq d_2 \geq \dots \geq d_{|H|}$ and d_h denote the degree of honest worker h.

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

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 M_E. Furthermore, the adversary uses its last identity a_k to label opposite the true label for 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 $\mathcal{T}_{cs}$ is the entire set of tasks $\mathcal{T}$. The bipartite graph $\mathcal{B}^{cs}$ decomposes into two disjoint bipartite graphs: bipartite graph $\mathcal{B}_E$ from H to E and semi-matching M_F from A to F that represents the adversary labeling strategy (it is a semi-matching because there is exactly one adversary labeling on 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 $\mathcal{B}_E$ and M_F. Since M_E is the optimal semi-matching on $\mathcal{B}_E$ and M_F is already a semi-matching by construction, the optimal semi-matching of $\mathcal{B}^{cs}$ is the disjoint union of M_E 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 atmost 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 on the entire task set and not just the conflict set $\mathcal{T}_{cs}$.

However, the bounds provided still hold as a result of lemma 21 above. Also, we assume that the adversary labeling strategy is always a semi-matching, i.e. there is atmost 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 22 *For any adversary labeling strategy, let $\mathcal{B}^{cs}(E)$ denote the bipartite graph $\mathcal{B}^{cs}$ restricted to all the workers W on the left and "true tasks" E on the right, and M be the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}$. Further, let $M(E) \subset M$ be the optimal semi-matching restricted to task copies E. Then, $M(E)$ is an optimal semi-matching for the sub-graph $\mathcal{B}^{cs}(E)$.*

Proof Suppose the statement is not true and let $N(E)$ denote the optimal semi-matching on $\mathcal{B}^{cs}(E)$. We use $d_w(K)$ to denote the degree of worker w in a semi-matching K. Note that, $d_a(N(E)) \le d_a(M(E)) \le d_a(M)$ for all adversaries $a \in A$. For the adversaries who did not agree with any honest worker, they will have degrees 0 in the semi-matchings $N(E)$

and $M(E)$ but the inequality is still satisfied. Now, since $N(E)$ is an optimal semi-matching and $M(E)$ is not, we have that

$$\begin{aligned}
&cost(N(E)) < cost(M(E)) \Rightarrow \\
&\sum_{h\in H} d_h(N(E))^2 + \sum_{a\in A} d_a(N(E))^2 < \sum_{h\in H} d_h(M(E))^2 + \sum_{a\in A} d_a(M(E))^2
\end{aligned}$$

Now, consider the semi-matching N on $\mathcal{B}^{cs}$ where we start with the semi-matching $N(E)$ and then map the adversaries A to the tasks in F to which they were assigned in the original optimal semi-matching M. Now, we claim that $cost(N) < cost(M)$ which will be a contradiction since M was assumed to be an optimal semi-matching on $\mathcal{B}^{cs}$.

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

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

We are now ready to prove the result for part (a).

Part 2a. Adversaries only provide incorrect responses or atmost 1 adversary provides correct responses

We start with the case when adversaries only provide incorrect responses.

Lemma 23 *Suppose that the adversaries never agree with the honest workers. 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 atmost $\sum_{i=1}^{k} b_i$.*

Proof It follows from the assumption that adversaries never agree with the honest workers that there are no cross-edges between 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 subgraph from the adversaries A to the task copies F. This further means that the semi-matching M is a disjoint union of semi-matchings on $\mathcal{B}(E)$ and $\mathcal{B}(F)$. Let the semi-matchings on the sub-graphs be termed as $M(E)$ and $M(F)$ respectively. Further, let $T \subseteq \mathcal{T}$ 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 reputations of the workers. We claim that $|T| \leq \sum_{i=1}^{k} b_i$. 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 $t \in \mathcal{T}$ such that h is mapped to the true copy of t in $M(E)$, a is mapped to the false 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 matching between $\bar{A}$ 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 sub-graphs by lemma 20 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 $H_1 = \mathrm{Img}(A_1)$ be the degenerate component. The number of tasks that adversaries in A_1 affect is bounded above by $\sum_{h \in \mathrm{Img}(A_1)} b_h$. Similarly, for $A_2 = \bar{A} \setminus A_1$, we can match each adversary to a *distinct* honest worker whose degree is greater than or equal to the degree of the adversary. 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 $|T|$ is atmost $\sum_{i=1}^{k} b_i$ if the adversaries only disagree with the honest workers. ■

Since the above lemma is true for *any* choice of semi-matching M, it is true 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 disagree with the honest workers.

Now, consider the case when there is exactly 1 adversary that agrees with the honest workers and all other adversaries only disagree. Let M be the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}$ resulting from such an adversary strategy and let a denote the adversary who agrees with the honest workers. Observe that we can apply the same argument in lemma 23 above to get a upper bound on the number of affected tasks that the adversaries "win" against the honest workers. Let T_1 denote the set of these tasks. In the proof of the lemma above, 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. In the other scenario, when the degenerate component is non-empty, we have a total of at most $k-1$ honest workers on the right and we bound T_1 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 a "loses" against *other adversaries*. The tasks that we might be missing out on correspond exactly to the task copies in E that the adversary a is mapped to in the optimal semi-matching M. Specifically, let $M(E)$ denote the semi-matching M restricted to just the true task copies E. Then it follows that we can bound the number of affected tasks by $|T_1|+d_a(M(E))$ where $d_a(M(E))$ denotes the degree of a in $M(E)$.

Next observe that in both cases, we have bounded the number of affected tasks by the sum of the degrees of some k workers in the semi-matching M restricted to workers $H \cup \{a\}$ on the left and tasks E on the right, i.e. in the semi-matching $M(E)$. Lemma 22 tells us that $M(E)$ is in fact, the optimal semi-matching on the subgraph from workers $H \cup \{a\}$ to task copies E. Finally, lemma 21 implies that this sum is atmost $\sum_{i=1}^{k} d_i$ (by starting with M_H as a feasible semi-matching) and the bound follows.

Next, we prove the result for part (b).

Part 2b. Adversaries can provide arbitrary responses

Finally, consider the general case when any number of adversaries can agree with the honest workers. We further suppose that $|H| \geq 2*|A|$, otherwise the upper bound U below becomes $\sum_{i=1}^{|H|} d_i$, which is the set of all tasks $\mathcal{T}$ and is a trivial upper bound.

First recall that lemma 23 was applicable to any semi-matching and in fact, we can use the same argument even when the adversaries agree with the honest workers. Formally, consider an arbitrary strategy resulting in an optimal semi-matching M on $\mathcal{B}^{cs}$. Let $M(E)$ denote the semi-matching M restricted to just the true task copies E. Suppose that the set of affected tasks T under this adversary strategy is such that $T = T_H \cup T_A$ where T_H are the tasks that the adversaries "win" against the honest workers and T_A 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 23 to bound $|T_H|$ by the sum of the degrees of the top k honest workers in the optimal semi-matching M. Further, we can bound T_A by the sum of the degrees of the adversaries in the semi-matching $M(E)$.

Let $A_H \subseteq A$ denote the set of adversaries that have non-zero degrees in semi-matching $M(E)$, i.e. they are mapped to some task in E in the optimal semi-matching M on $\mathcal{B}^{cs}$. The above sequence of claims implies that we can bound the number of affected tasks $|T|$ by the sum of the degrees of the top $s = k + |A_H|$ workers in the semi-matching $M(E)$. Now, we claim that this itself is upper bounded by the sum of the degrees of the top j honest workers in the optimal semi-matching M_H on the original honest worker-task assignment sub-graph $\mathcal{B}_H$. To see this, start with M_H as a feasible semi-matching from workers $H \cup A_H$ to task copies E. Lemma 21 tells us that the sum of the degrees of the top j workers in the optimal semi-matching on the subgraph $\mathcal{B}^{cs}(E)$ is atmost the sum of the degrees of the top j honest workers in M_H. Further, lemma 22 tells us that the optimal semi-matching on the sub-graph from workers $H \cup A_H$ to task copies E is precisely the semi-matching $M(E)$. This shows that we can bound the number of affected tasks by $\sum_{i=1}^{j} d_i$. Finally, note that $|A_H| \leq k \Rightarrow j \leq 2k$ and hence, we can bound the total number of affected tasks by $\sum_{i=1}^{2k} d_i$.

A.6 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 24 *Let M and M' be two optimal semi-matchings on a bipartite graph $G = (V_1, V_2, E)$ with $|U| = n$ 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 L^p norm of the degree-vector for any $p \geq 1$ (Section 3.4 in [OPT]). Hence, we have that $d_i = d'_i \; \forall i$. ∎

A.7 Relation between Lower bound L and Optimal semi-matching degrees

We prove here the relationship between the lower bound L in theorem 12 and the honest worker degrees $d_1, d_2, \ldots, d_{|H|}$ in the optimal semi-matching on the honest worker-task assignment graph $\mathcal{B}_H$.

Lemma 25 *Let $d_1 > d_2 > \cdots > d_{|H|}$ denote the degrees of the honest workers in the optimal semi-matching M_H on $\mathcal{B}_H$. Then the lower bound L in theorem 12 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 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 M_H is an optimal semi-matching. Specifically, Theorem 3.1 in [OPT] 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 $\mathcal{B}_H$ 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 [OPT] for 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. For the sake of completeness, we describe the setup in Raykar and Yu (2012), also known as the "two-coin" model. For each worker w_i, her accuracy is modeled separately for positive and negative tasks. 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 $\mathbf{\Theta} = [\left\{(\alpha_i, \beta_i) \,\Big|\, i \in [n]\right\}, \gamma]$ 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:

$$\log Pr[\mathcal{L}, \boldsymbol{y} \,\Big|\, \mathbf{\Theta}] = \sum_{j=1}^{m} \Big(y_j \log(a_j \gamma) + (1-y_j) \log(1-\gamma) b_j \Big) \tag{5}$$

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

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

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

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

- **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 $\mathbb{E}\left\{\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \boldsymbol{\Theta}^{(k)}]\right\}$, which can be shown to be a lower bound on the data log-likelihood (eq 5). 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 $\Pr[\mathcal{L} \mid \boldsymbol{\Theta}]$. To initialize the EM algorithm, we use the majority estimate $\gamma_j^{(0)} = \frac{\sum_{i=1}^n \mathbf{1}[\mathcal{L}_{ij} = +1]}{n}$.

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 p_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.

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 $\hat{y}_j = \mathrm{sign}(\mathrm{m_j})$

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 $m^{(0)}_{i\to j}$ 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 $m^{(k)}_{j\to i} = \frac{1}{deg_{\mathcal{B}}(t_j)} \sum_{i' \neq i} \mathcal{L}_{ij} m^{(k-1)}_{i\to j}$
- For all $(w_i, t_j) \in E$, update $m^{(k)}_{i\to j} = \frac{1}{deg_{\mathcal{B}}(w_i)} \sum_{j' \neq j} \mathcal{L}_{ij} m^{(k)}_{j\to i}$

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

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

Other benchmarks. In addition to the benchmarks discussed in the main text, we also implemented two more methods: the SPECTRAL EM algorithm by Zhang et al. [SPEM] and the regularized minimax conditional entropy (MMCE) approach by Zhou et al. [MMCE]. The following table reports the results on the real world datasets – for the SPECTRAL EM algorithm we obtain improvements throughout, whereas for MMCE, we boost the accuracy in 3 cases. This shows that our approach works with a wide variety of benchmark label aggregation algorithms.

Dataset	SPECTRAL EM			MMCE		
	BASE	SOFT	HARD’	BASE	SOFT	HARD’
rte	91.9	92.1(7)	**92.6(3)**	92.5	92.9(9)	**93.2(3)**
temp	93.9	93.9	**94.6(3)**	94.4	94.4	94.4
stage2	74.3	75.4(1)	**80.7(3)*****	57.0	58.8 (5)	**73.3(10)*****
task2	64.2	64.2	**67.8(10)*****	66.1	66.1	**66.9(10)**
tweets	69.6	69.8(4)	**73.3(1)*****	72.7	72.7	72.7
aggregate	80.0	80.3	**82.3**	82.8	84.2	**85.5**

Simulation Details. Here we discuss how we imposed the degree bias on adversaries in the Preferential Attachment scenario. 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

[EXP] Salil P Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science: Vol. 7: No. 1-3*, pp 1-336, 2012.

[OPT] 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.

[SPEM] 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.

[MMCE] 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.