Adaptive Task Assignment for Crowdsourced Classification

Chien-Ju Ho, Shahin Jabbari CJHO@CS.UCLA.EDU, SHAHIN@CS.UCLA.EDU
University of California, Los Angeles

Jennifer Wortman Vaughan JENN@MICROSOFT.COM
Microsoft Research, New York City and University of California, Los Angeles

Abstract

Crowdsourcing markets have gained popularity as a tool for inexpensively collecting data from diverse populations of workers. Classification tasks, in which workers provide labels (such as "offensive" or "not offensive") for instances (such as "websites"), are among the most common tasks posted, but due to human error and the prevalence of spam, the labels collected are often noisy. This problem is typically addressed by collecting labels for each instance from multiple workers and combining them in a clever way, but the question of how to choose which tasks to assign to each worker is often overlooked. We investigate the problem of task assignment and label inference for heterogeneous classification tasks. By applying online primal-dual techniques, we derive a provably near-optimal adaptive assignment algorithm. We show that adaptively assigning workers to tasks can lead to more accurate predictions at a lower cost when the available workers are diverse.

1. Introduction

Crowdsourcing markets provide a platform for inexpensively harnessing human computation power to solve tasks that are notoriously difficult for computers. In a typical crowdsourcing market, such as Amazon Mechanical Turk, registered users may post their own "microtasks" which are completed by workers in exchange for a small payment, usually around ten cents. A microtask may involve, for example, verifying the phone number of a business, determining whether or not an image contains a tree, or determining (subjectively) whether or not a particular website is offensive.

The availability of diverse workers willing to complete tasks inexpensively makes crowdsourcing markets appealing as tools for collecting data (Wah et al., 2011). Classification tasks, in which workers are asked to provide a binary label for an instance, are among the most common tasks posted (Ipeirotis, 2010). Unfortunately, due to a mix of human error, carelessness, and fraud — the existence of spammy workers on Mechanical Turk is widely acknowledged — the data collected is often noisy (Kittur et al., 2008; Wais et al., 2010). For classification tasks, this problem can be overcome by collecting labels for each instance from multiple workers and combining these to infer the true label. Indeed, much recent research has focused on developing algorithms for combining labels from heterogeneous labelers (Dekel & Shamir, 2009; Ipeirotis et al., 2010). However, this research has typically focused on the inference problem, sidestepping the question of how to assign workers to tasks by assuming that the learner has no control over the assignment. One exception is the work of Karger et al. (2011a;b), who focus on the situation in which all tasks are homogeneous (i.e., equally difficult and not requiring specialized skills), in which case they show that it is not possible to do better than using a random assignment.

One might expect the assignment to matter more when the tasks are heterogeneous. Classifying images of dogs versus images of cats is likely easier for the average worker than classifying images of Welsh Terriers versus images of Airedale Terriers. It might be necessary to assign more workers to tasks of the latter type to produce high confidence labels. The assignment can also be important when tasks require specialized skills. A worker who knows little about dogs may not be able to produce high quality labels for the Terrier task, but may have skills that are applicable elsewhere.

We investigate the problem of task assignment and label inference for heterogeneous classification tasks. In

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

our model, a task requester has a set of tasks, each of which consists of an instance for which he would like to infer a binary label. Workers arrive online. The learner must decide which tasks to assign to each worker, and then use the noisy labels produced by the workers to infer the true label for each task. The goal of the learner is to output a set of labels with sufficiently low error while requesting as few labels from workers as possible. Building on online primal-dual methods (Buchbinder & Naor, 2005), we propose an exploration-exploitation algorithm that is provably competitive with an optimal offline algorithm that has knowledge of the sequence of workers and their skills in advance. We then evaluate this algorithm in a variety of experiments on synthetic data and show that adaptively allocating tasks helps when the worker distribution is diverse or the tasks are heterogeneous.

2. Related Work

Our research is mostly closely related to that of Karger et al. (2011a;b) and Ho & Vaughan (2012). Karger et al. introduced a model in which a requester has a set of homogeneous labeling tasks he must assign to workers who arrive online. They proposed an assignment algorithm based on random graph generation and a message-passing inference algorithm inspired by belief propagation, and showed that their technique is order-optimal in terms of labeling budget. In particular, let p_j be the probability that worker j completes any given task correctly and $q = \mathbb{E}[(2p_j - 1)^2]$, where the expectation is over the choice of a random worker j. They proved that their algorithm requires $O((1/q)\log(1/\epsilon))$ labels per task to achieve error less than ϵ in the limit as the numbers of tasks and workers go to infinity. They also showed that adaptively assigning tasks does not help in their setting, in that $\Omega((1/q)\log(1/\epsilon))$ labels are still needed in general.

We generalize this model to allow heterogeneous tasks, so that the probability that worker j completes a task correctly may depend on the particular task. In this generalized setting, assigning tasks adaptively can provide an advantage both in theory and in practice.

Our techniques build on the online primal-dual framework, which has been used to analyze online optimization problems ranging from the adwords problem (Buchbinder et al., 2007; Devanur et al., 2011) to network optimization (Alon et al., 2004) and paging (Bansal et al., 2007). Ho & Vaughan (2012) were the first to apply this framework to crowdsourcing. In their model, a requester has a fixed set of tasks of different types, each of which must be completed exactly once. Each worker has an unknown skill level for each type of task, with workers of higher skill levels producing higher quality work on average. Workers arrive online, and the learner must assign each worker to a single task upon arrival. When the worker completes the task, the learner immediately receives a reward, and thus also a noisy signal of the worker's skill level for tasks of that type. Workers arrive repeatedly and are identifiable, so the learner can form estimates of the workers' skill levels over time. The goal is to maximize the sum of requester rewards. Ho & Vaughan provide an algorithm based on the online primal-dual framework and prove that this algorithm is competitive with respect to the optimal offline algorithm that has access to the unknown skill levels of each worker.

Our model differs from that of Ho & Vaughan in several key ways. Their analysis depends heavily on the assumption that the requester can evaluate the quality of completed work immediately (i.e., learn his reward on each time step), which is unrealistic in many settings, including the labeling task we consider here; if the requester could quickly verify the accuracy of labels, he wouldn't need the workers' labels in the first place. In their model, each task may be assigned to a worker only once. In ours, repeated labeling is necessary since there would be no way to estimate worker quality without it. These differences require a different problem formulation and novel analysis techniques.

Repeated labeling has received considerable empirical attention, dating back to the EM-based algorithm of Dawid & Skene (1979). Sheng et al. (2008) considered a setting in which every worker is correct on every task with the same probability, and empirically evaluated how much repeated labeling helps. Ipeirotis et al. (2010) extended this idea to heterogeneous workers and provided an algorithm to simultaneously estimate workers' quality and true task labels. More recently, there has been work showing that label inference can be improved by first estimating parameters of the structure underlying the labeling process using techniques such as Bayesian learning (Welinder et al., 2010), minimax entropy (Zhou et al., 2012), and variational inference (Liu et al., 2012).

On the theoretical side, there have been several results on learning a binary classifier using labeled data contributed by multiple teachers, each of which labels instances according to his own fixed labeling function (Crammer et al., 2005; 2008; Dekel & Shamir, 2009). These require PAC-style assumptions and focus on filtering out low quality workers. Tran-Thanh et al. (2012) used ideas from the multi-armed bandit literature to assign tasks. Bandit ideas cannot be applied in our setting without further assumptions since the reward corresponding to an assignment depends on whether the worker's label is correct, which cannot be inferred until the task has been assigned to others.

Ghosh et al. (2011) studied a model similar to that of Karger et al., also with homogeneous tasks, and used eigenvalue decomposition to estimate each worker's quality. Their bounds depend on a quantity essentially identical to the quantity q defined above, which they refer to as the population's *average competence*. A similar quantity plays a role in our analysis.

3. The Model

In our model, a task requester has a set of n tasks, indexed $1, \cdots, n$. Each task is a binary classification problem. The true label of task i, denoted ℓ_i, is either 1 or -1, and is unknown to the requester.

Workers arrive online. When worker j arrives, she announces the maximum number of tasks that she is willing to complete, her *capacity*, M_j. No other information is known about each worker when she arrives.

Each worker j has a *skill level*, $p_{i,j} \in [0,1]$, for each task i. If the algorithm assigns worker j to task i, the worker will produce a label $\ell_{i,j}$ such that $\ell_{i,j} = \ell_i$ with probability $p_{i,j}$ and $\ell_{i,j} = -\ell_i$ with probability $1-p_{i,j}$, independent of all other labels. The algorithm may assign worker j up to M_j tasks, and may observe her output on each task before deciding whether to assign her to another or move on, but once the algorithm moves on, it cannot access the worker again. This is meant to reflect that crowdsourced workers are neither identifiable nor persistent, so we cannot hope to identify and later reuse highly skilled workers.

Several of our results depend on the quantity $q_{i,j} = (2p_{i,j} - 1)^2$. Intuitively, when this quantity is close to 1, the label of worker j on task i will be informative; when it is close to 0, the label will be random noise.

To model the fact that the requester cannot wait arbitrarily long, we assume that he can only assign tasks to the first m workers who arrive, for some known m. We therefore index workers $1, \cdots, m$. Later we consider an additional γm workers who are used for exploration.

In addition to assigning tasks to workers, the learning algorithm must produce a final estimate $\hat{\ell}_i$ for the label ℓ_i of each task i based on the labels provided by the workers. The goal of the learner is to produce estimates that are correct with high probability while querying workers for as few labels as possible.

Task structure: A clever learning algorithm should infer the worker skill levels $p_{i,j}$ and assign workers to tasks at which they excel. If the skills are arbitrary, then the learner cannot infer them without assigning every worker to every task. Therefore, it is necessary to assume that the $p_{i,j}$ values exhibit some structure. Karger et al. (2011a;b) assume that all tasks are identical, i.e., $p_{i,j} = p_{i',j}$ for all j and all i and i'. We consider a more general setting in which the tasks can be divided into T *types*, and assume only that $p_{i,j} = p_{i',j}$ if i and i' are of the same type.

Gold standard tasks: As is common in the literature (Oleson et al., 2011), we assume that the learner has access to "gold standard" tasks of each task type.[1] These are instances for which the learner knows the true label a priori. They can be assigned in order to estimate the $p_{i,j}$ values. Of course the algorithm must pay for these "pure exploration" assignments.

Random permutation model: We analyze our algorithm in the random permutation model as in Devanur & Hayes (2009). The capacities M_j and skills $p_{i,j}$ of each worker j may be chosen adversarially, as long as the assumptions on task structure are satisfied. However, the arrival order is randomly permuted. Since only the order of workers is randomized, the offline optimal allocation is well-defined.

Competitive ratio: To evaluate our algorithm, we use the notion of *competitive ratio*, which is an upper bound on the ratio between the number of labels requested by the algorithm and the number requested by an *optimal offline algorithm* which has access to all worker capacities and skill levels, but must still assign enough workers to each task to obtain a high-confidence guess for the task's label. The optimal offline algorithm is discussed in Sections 4 and 5.

4. An Offline Problem

To gain intuition, we first consider a simplified offline version of our problem in which the learner is provided with a full description of the sequence of m workers who will arrive, including the skill levels $p_{i,j}$ and capacities M_j for all i and j. The learner must decide which tasks to assign to each worker and then infer the task labels. We discuss the inference problem first.

4.1. Aggregating Workers' Labels

Suppose that the learner has already assigned tasks to workers and observed the workers' labels for these tasks. How should the learner aggregate this information to infer the true label for each task?

[1] If gold standard tasks are not available, they can be created by assigning a small set of tasks to many workers.

We consider aggregation methods that take a weighted vote of the workers' labels. Fix a task i. Let J_i denote the set of workers assigned to this task. We consider methods that set $\hat{\ell}_i = \text{sign}(\sum_{j \in J_i} w_{i,j} \ell_{i,j})$ for some set of weights $\{w_{i,j}\}$. The following lemma shows that this technique with weights $w_{i,j} = 2p_{i,j} - 1$ is guaranteed to achieve a low error if enough high quality workers are queried. Recall that $q_{i,j} = (2p_{i,j} - 1)^2$.

Lemma 1. *Let $\hat{\ell}_i = \text{sign}(\sum_{j \in J_i} w_{i,j} \ell_{i,j})$ for some set of weights $\{w_{i,j}\}$. Then $\hat{\ell}_i \neq \ell_i$ with probability at most $e^{-\frac{1}{2}(\sum_{j \in J_i} w_{i,j}(2p_{i,j}-1))^2 / \sum_{j \in J_i} w_{i,j}^2}$. This bound is minimized when $w_{i,j} \propto (2p_{i,j} - 1)$, in which case the probability that $\hat{\ell}_i \neq \ell_i$ is at most $e^{-\frac{1}{2}\sum_{j \in J_i} q_{i,j}}$.*

The proof, which uses a simple application of Hoeffding's inequality, is in the appendix.[2] This tells us that to guarantee that we make an error with probability less than ϵ on a task i, it is sufficient to select a set of labelers J_i such that $\Sigma_{j \in J_i} q_{i,j} \geq 2 \ln(1/\epsilon)$ and aggregate labels by setting $\hat{\ell}_i = \text{sign}(\sum_{j \in J_i} (2p_{i,j} - 1) \ell_{i,j})$.

One might ask if it is possible to guarantee an error of ϵ with fewer labels. In some cases, it is; if there exists an i and j such that $p_{i,j} = q_{i,j} = 1$, then one can achieve zero error with only a single label. However, in some cases this method is optimal. For this reason, we restrict our attention to algorithms that query subsets J_i such that $\Sigma_{j \in J_i} q_{i,j} \geq 2 \ln(1/\epsilon)$. We use the shorthand $C_\epsilon = 2 \ln(1/\epsilon)$.

4.2. Integer Programming Formulation

There is a significant benefit that comes from restricting attention to algorithms of the form described above. Let $y_{i,j}$ be a variable that is 1 if task i is assigned to worker j and 0 otherwise. The requirement that $\sum_{j \in J_i} q_{i,j} \geq C_\epsilon$ can be expressed as a linear constraint of these variables. This would not be possible using unweighted majority voting to aggregate labels; weighting by $2p_{i,j} - 1$ is key. This allows us to express the optimal offline assignment strategy as an integer linear program (IP), with variables $y_{i,j}$ for each (i, j):

$$\begin{aligned} \min \;& \Sigma_{i=1}^n \Sigma_{j=1}^m y_{i,j} \\ \text{s.t. } & \Sigma_{i=1}^n y_{i,j} \leq M_j \;\; \forall j && (1) \\ & \Sigma_{j=1}^m q_{i,j} y_{i,j} \geq C_\epsilon \;\; \forall i && (2) \\ & y_{i,j} \in \{0, 1\} \;\; \forall (i,j). && (3) \end{aligned}$$

Constraint (1) guarantees that worker j does not exceed her capacity. Constraint (2) guarantees that aggregation will produce the correct label of each task with high probability. Constraint (3) implies that a task is either assigned to a worker or not.

Note that there may not exist a feasible solution to this IP, in which case it would not be possible to guarantee a probability of error less than ϵ for all tasks using weighted majority voting. For most of this paper, we assume a feasible solution exists; the case in which one does not is discussed in Section 5.1.

For computational reasons, instead of working directly with this IP, we will work with a linear programming relaxation obtained by replacing the last constraint with $0 \leq y_{i,j} \leq 1 \;\forall (i, j)$; we will see below that this does not impact the solution too much.

4.3. Working with the Dual

Solving the linear program described above requires knowing the values $q_{i,j}$ for the full sequence of workers j up front. When we move to the online setting, it will be more convenient to work with the dual of the relaxed linear program, which can be written as follows, with dual variables x_i, z_j, and $t_{i,j}$ for all (i, j):

$$\begin{aligned} \max \;& C_\epsilon \Sigma_{i=1}^n x_i - \Sigma_{j=1}^m M_j z_j - \Sigma_{i=1}^n \Sigma_{j=1}^m t_{i,j} \\ \text{s.t. } & 1 - q_{i,j} x_i + z_j + t_{i,j} \geq 0 \;\; \forall (i,j) \\ & x_i, z_j, t_{i,j} \geq 0 \;\; \forall (i,j). \end{aligned}$$

We refer to x_i as the *task weight* for i, and define the *task value* of worker j on task i as $v_{i,j} = q_{i,j} x_i - 1$.

Suppose that we were given access to the task weights x_i for each task i and the values $q_{i,j}$. (We will discuss how to approximate these values later.) Then we could use the following algorithm to approximate the optimal primal solution. Note that to run this algorithm, it is not necessary to have access to all $q_{i,j}$ values at once; we only need information about worker j when it comes time to assign tasks to this worker. This is the advantage of working with the dual.

Algorithm 1 Primal Approximation Algorithm

Input: Values x_i and $q_{i,j}$ for all (i, j)
For every worker $j \in \{1, \ldots, m\}$, compute the task values, $v_{i,j} = q_{i,j} x_i - 1$, for all tasks i. Let n_j be the number of tasks i such that $v_{i,j} \geq 0$. If $n_j \leq M_j$, then set $y_{ij} \leftarrow 1$ for all n_j tasks with non-negative task value. Otherwise, set $y_{i,j} \leftarrow 1$ for the M_j tasks with highest task value. Set $y_{i,j} \leftarrow 0$ for all other tasks.

The following theorem shows that this algorithm produces a near-optimal primal solution to our original IP when given as input the optimal dual solution for the relaxed LP. The condition that $q_{i,j} x_i^* \neq q_{i',j} x_{i'}^*$ for all $i \neq i'$ is needed for technical reasons, but can be

[2] An appendix containing all omitted proofs and additional details can be found in the long version of this paper available on the authors' websites.

relaxed by adding small random perturbations to $q_{i,j}$ values as in Devanur et al. (2011).[3] For the rest of the paper, we assume that perturbations have been added and that the condition above holds. Call this the "perturbation assumption." In our final algorithm, we will perturb our estimates of the $q_{i,j}$ values for this reason.

Theorem 1. *Let $\boldsymbol{y}^*$ be the primal optimal of the IP and $\boldsymbol{x}^*$ be the dual optimal of the relaxed formulation. Let $\boldsymbol{y}$ be the output of the Primal Approximation Algorithm given input $\boldsymbol{x}^*$ and the true values $\boldsymbol{q}$. Then $\boldsymbol{y}$ is feasible in the IP, and under the perturbation assumption, $\Sigma_{i=1}^n \Sigma_{j=1}^m y_{i,j} - \Sigma_{i=1}^n \Sigma_{j=1}^m y^*_{i,j} \le \min(m,n)$.*

The proof shows that the $y_{i,j}$ values assigned by the Primal Approximation Algorithm differ from the optimal solution of the relaxed LP for at most $\min(m,n)$ pairs (i,j), and that this implies the result.

5. Moving to the Online Setting

We have shown that, given access to $\mathbf{q}$ and the optimal task weights $\mathbf{x}^*$, the Primal Approximation Algorithm generates an assignment which is close to the optimal solution of the IP in Section 4.2. However, in the online problem that we initially set out to solve, these values are unknown. In this section, we provide methods for estimating these quantities and incorporate these into an algorithm for the online problem.

Our online algorithm combines two varieties of exploration. First, we use exploration to estimate the optimal task weights $\mathbf{x}^*$. To do this, we hire an additional γm workers on top of the m workers we originally planned to hire, for some $\gamma > 0$, and "observe" their $q_{i,j}$ values. (We will actually only estimate these values; see below.) Then, by treating these γm workers as a random sample of the population (which they effectively are under the random permutation model), we can apply online primal-dual methods and obtain estimates of the optimal task weights. These estimates can then be fed to the Primal Approximation Algorithm in order to determine assignments for the remaining m workers, as described in Section 5.1.

The second variety is used to estimate workers' skill levels. Each time a new worker arrives (including the γm extras), we require her to complete a set of gold standard tasks of each task type. Based on the labels she provides, we estimate her skill levels $p_{i,j}$ and use these to estimate the $q_{i,j}$ values. The impact of these estimates on performance is discussed in Section 5.2.

[3] Adding noise will introduce an error, but this error can be made arbitrarily small when C_ϵ is large. To simplify presentation, we do no include the error in our discussion.

If we require each worker to complete s gold standard tasks, and we hire an extra γm workers, we need to pay for an extra $(1+\gamma)ms$ assignments beyond those made by the Primal Approximation Algorithm. We precisely quantify how the number of assignments compares with the offline optimal in Section 5.3.

5.1. Estimating the Task Weights

In this section, we focus on the estimation of task weights in a simplified setting in which we can observe the quality of each worker as she arrives. To estimate the task weights, we borrow an idea from the literature on the online primal-dual framework. We use an initial sampling phase in which we hire γm workers in addition to the primary m workers, for some γ. We observe their skill levels and treat the distribution over skills of the sampled workers as an estimate of the distribution of skills of the m primary workers. Given the $q_{i,j}$ values from the sampled γm workers, we can solve an alternative linear programming problem, which is the same as our relaxed offline linear programming problem, except that m is replaced by γm and C_ϵ is replaced by γC_ϵ. Let $\hat{\mathbf{x}}^*$ be the optimal task weights in this "sampled LP" problem. We show that if ϵ is small enough, running the Primal Approximation Algorithm using $\hat{\mathbf{x}}^*$ and $\mathbf{q}$ yields a near-optimal solution, with a number of assignments close to optimal, and a prediction error close to ϵ after aggregation.

Theorem 2. *For any $\epsilon, \delta \in (0, 1/2)$, for any $\gamma = \ell/m$ with $\ell \in \{1, 2, \cdots, m\}$ and $\gamma \in [1/C_\epsilon, 1]$, let $\hat{\boldsymbol{y}}^{s,*}$ and $\hat{\boldsymbol{x}}^*$ be the primal and dual optimal solutions of the sampled LP with parameters ϵ and γ. Let $\hat{\boldsymbol{y}}^*$ be the output of the Primal Approximation Algorithm with inputs $\hat{\boldsymbol{x}}^*$ and $\boldsymbol{q}$, and let $\bar{\boldsymbol{y}}^*$ be the optimal assignment of the relaxed offline formulation with parameter ϵ. Let $q_{min} = \min_{(i,j):\hat{y}^{s,*}_{i,j}>0} q_{i,j}$. Then under the perturbation assumption, with probability at least $1-\delta$,*

$$\sum_{i=1}^n \sum_{j=1}^m \hat{y}^*_{i,j} \le \left(1 + \frac{\min(m,n)}{q_{min} n C_\epsilon} + \frac{35 \ln(2/\delta)}{q_{min}\sqrt{\gamma C_\epsilon}}\right) \sum_{i=1}^n \sum_{j=1}^m \bar{y}^*_{i,j}.$$

If the labels collected from the resulting assignment are used to estimate the task labels via weighted majority voting, the probability that any given task label is predicted incorrectly is no more than $\epsilon^{1-6\ln(2/\delta)/\sqrt{\gamma C_\epsilon}}$.

The requirement that $\gamma \ge 1/C_\epsilon$ stems from the fact that if C_ϵ is small, the total number of assignments will also be small, and the quality of the assignment is more sensitive to estimation errors. If C_ϵ is large, small estimation errors effect the assignment less and we can set the sampling ratio to a smaller value.

In the proof, we show that the gap between the objectives of the primal solution generated by the Primal Approximation Algorithm using $\hat{\mathbf{x}}^*$ and $\mathbf{q}$ and the corresponding dual solution is exactly the summation of $\hat{x}_i^*(C_\epsilon - \Sigma_{j=1}^m q_{i,j}\hat{y}_{i,j}^*)$ over all tasks i, which is small if enough workers are sampled. By weak duality, the optimal number of assignments is between the primal and the dual objectives, so the primal solution output by the algorithm must be near-optimal.

A note on feasibility: We have implicitly assumed that the sampled LP is feasible. In practice, it may not be, or even if it is, there may exist tasks i such that $\min_{j:\hat{y}_{i,j}^{s,*}>0} q_{i,j}$ is very small, leading to a small value of q_{min}. If either of these things happen, the task requester may want to discard some of the tasks or lower his desired error, solve the sampled LP with these modified constraints, and continue from there, as there is no way to guarantee low error on all tasks.

5.2. Using Estimates of Skill Levels

We now discuss the effect of estimating worker skills. Given observations of the gold standard tasks of type τ that worker j completed, we can estimate $p_{i,j}$ for any task i of type τ as the fraction of these tasks she labeled correctly. The following lemma, follows from a straightforward application of the Hoeffding bound; we state it here as it will be useful, but omit the proof.

Lemma 2. *For any worker j, for any task type τ, and for any $t, \delta \in (0,1)$, suppose that worker j labels $\ln(2/\delta)/(2t^2)$ gold standard tasks of type τ. Then with probability at least $1-\delta$, for all tasks i of type τ, if we set $\hat{p}_{i,j}$ to the fraction of gold standard tasks of type τ answered correctly then $|p_{i,j} - \hat{p}_{i,j}| \le t$.*

This estimate of $p_{i,j}$ can then be used to derive an estimate for $q_{i,j}$, with error bounded as follows.

Lemma 3. *For any worker j and task i, if $\hat{p}_{i,j}$ is an estimate of $p_{i,j}$ such that $|p_{i,j} - \hat{p}_{i,j}| \le t$, and $\hat{q}_{i,j}$ is set to $(2\hat{p}_{i,j} - 1)^2$, then $|q_{i,j} - \hat{q}_{i,j}| \le 4t$.*

Of course the use of estimated values impacts performance. Consider the offline problem discussed in the previous section. One might hope that if we applied the Primal Approximation Algorithm using $\hat{\mathbf{q}}$, the number of assignments would be close to the number made using $\mathbf{q}$. Unfortunately, this is not true. Consider this toy example. Let $q_{i,1} = q_{i,2} = q_{i,3} = 1$ for all i, $q_{i,j} = 10^{-4}$ for all i and $j > 3$, and $M_j = n$ for all j. Set $\epsilon = 0.224$ so that $C_\epsilon \approx 3$. In the optimal solution, each task i should be assigned only to workers $1, 2$, and 3. If we underestimate the $q_{i,j}$ values, we could end up assigning each task to many more workers. This can be made arbitrarily bad.

To address this, instead of solving the relaxed offline formulation directly, we consider an alternative LP which is identical to the relaxed offline formulation, except that $\mathbf{q}$ is replaced with $\hat{\mathbf{q}}$ and C_ϵ is replaced with a smaller value $C_{\epsilon'}$ (corresponding to a higher allowable error ϵ'). We call this *the approximated LP*. We show that, if ϵ' is chosen properly, we can guarantee the optimal solution in the relaxed offline formulation is feasible in the approximated LP, so the optimal solution of the approximated LP will yield an assignment with fewer tasks assigned to workers than the optimal solution of the relaxed offline formulation, even though it is based on estimations.

To set ϵ', we assume the requester has a rough idea of how hard the tasks are and how inaccurate his estimates of worker skills are. The latter can be achieved by applying Lemma 2 and the union bound to find a value of t such that $|p_{i,j} - \hat{p}_{i,j}| \le t$ for all (i,j) pairs with high probability, and setting each $\hat{q}_{i,j} = (2\hat{p}_{i,j} - 1)^2$ as in Lemma 3. For the former, let $\bar{\mathbf{y}}^*$ be the optimal solution of the relaxed offline formulation. Define $\bar{q}_i^* = \Sigma_{j=1}^m q_{i,j}\bar{y}_{i,j}^* / \Sigma_{j=1}^m \bar{y}_{i,j}^*$. We assume that the requester can produce a value $\bar{q}_{min}^*$ such that $\bar{q}_{min}^* \le \bar{q}_i^*$ for all i and then set $C_{\epsilon'} = 2\ln(1/\epsilon')$ where $\epsilon' = \epsilon^{1-4t/\bar{q}_{min}^*}$. If the requester doesn't have much information, he can conservatively set $\bar{q}_{min}^*$ much smaller than $\min_i\{\bar{q}_i^*\}$, but will both need more accurate estimates of $p_{i,j}$ and sacrifice some prediction accuracy.

Theorem 3. *Assume that we have access to a value $\bar{q}_{min}^*$ such that $\bar{q}_{min}^* \le \bar{q}_i^*$ for all i and values $\hat{p}_{i,j}$ such that $|p_{i,j} - \hat{p}_{i,j}| \le t$ for all (i,j) pairs for any known value $t < \bar{q}_{min}^*/4$. Then for any $\epsilon > 0$, the optimal solution of the approximated LP with parameter $\epsilon' = \epsilon^{1-4t/\bar{q}_{min}^*}$ and skill levels $\hat{q}_{i,j} = (2\hat{p}_{i,j} - 1)^2$ is no bigger than the optimal solution of the relaxed offline formulation with parameter ϵ and skill levels $q_{i,j}$.*

Of course this guarantee is not free. We pay the price of decreased prediction accuracy since we are using ϵ' in place of ϵ. We also pay when it comes time to aggregate the workers' labels, since we must now use $\hat{\mathbf{q}}$ in place of $\mathbf{q}$ when applying the weighted majority voting method described in Section 4.1. This is quantified in the following theorem. Note that this theorem applies to *any* feasible integer solution of the approximated LP and therefore also the best integer solution.

Theorem 4. *Assume again that we have access to a value $\bar{q}_{min}^*$ such that $\bar{q}_{min}^* \le \bar{q}_i^*$ for all i and values $\hat{p}_{i,j}$ such that $|p_{i,j} - \hat{p}_{i,j}| \le t$ for all (i,j) pairs for any known value $t < \bar{q}_{min}^*/4$. For any $\epsilon > 0$, let $\boldsymbol{y}$ be any feasible integer assignment of the approximated LP with parameter $\epsilon' = \epsilon^{1-4t/\bar{q}_{min}^*}$ and skill levels $\hat{q}_{i,j} = (2\hat{p}_{i,j} - 1)^2$. Let $J_i = \{j : y_{i,j} = 1\}$ denote the* *set of workers that are assigned to task i according to $\boldsymbol{y}$, and define $\hat{q}_i = \Sigma_{j \in J_i} q_{i,j} / |J_i|$. If the tasks are assigned according to $\boldsymbol{y}$ and the results aggregated using weighted majority voting with weights $w_{i,j} = 2\hat{p}_{i,j} - 1$, the error probability of the predicted task label for task i is bounded by $\epsilon^{(1-4t/\bar{q}^*_{min})(1-4t/\hat{q}_i)}$.*

Theorem 4 tells us that if our estimates of the worker skills are accurate (i.e., if t is small), then our prediction error will be close to the error we would have achieved if we had known the true worker skills. How good the estimates need to be depends on the quality of the workers, as measured by $\bar{q}^*_{min}$ and $\hat{q}_i$. Intuitively, if $\bar{q}^*_{min}$ is small, there may exist some task i at which workers perform poorly in the optimal solution. In this case, the assignment will be very sensitive to the value of ϵ' chosen, and it will be necessary to set ϵ' larger to guarantee that the true optimal solution is feasible in the approximated LP. If $\hat{q}_i$ is small, then a small amount of error in estimated worker quality would dramatically change the weights used in the weighted majority voting aggregation scheme.

5.3. Putting it All Together

We have separately considered relaxations of the task assignment and label inference problem in which the optimal task weights or worker skill levels are already known. We now put all these pieces together, give a combined algorithm, and state our main theorem.

Algorithm 2 Main Algorithm

Input: Values $(\epsilon, \gamma, s$, and $\bar{q}^*_{min})$
Hire γm preliminary workers.
for each preliminary worker **do**
  Assign s gold standard tasks of each task type.
  Calculate $\hat{q}_{i,j}$ values as in Section 5.2 and perturb with a negligible amount of noise.
end for
Calculate $C_{\epsilon'}$ and solve the sampled LP with $\hat{\mathbf{q}}$ to obtain primal $\mathbf{y}^s$ and dual $\hat{\mathbf{x}}^*$ as in Section 5.1.
for each worker $j \in \{1, ..., m\}$ **do**
  Assign s gold standard tasks of each task type.
  Calculate $\hat{q}_{i,j}$ values as in Section 5.2 and perturb with a negligible amount of noise.
  Run the Primal Approximation Algorithm with inputs $\hat{\mathbf{x}}^*$ and (perturbed) $\hat{\mathbf{q}}$ to $\mathbf{y}$.
  Assign worker j to all tasks i with $y_{i,j} = 1$.
end for
Aggregate the workers' labels using weighted majority voting as in Section 4.1.

The complete algorithm is stated in Algorithm 2, and its performance guarantee is given below. Recall that T is the number of task types. Again, we assume that the optimization problems are feasible.

Theorem 5. *For any $\epsilon, \delta \in (0, 1/2)$, for any $\gamma = \ell/m$ for an $\ell \in \{1, 2, ..., m\}$ such that $\gamma \in [1/C_\epsilon, 1]$, assume we have access to a value $\bar{q}^*_{min}$ satisfying the condition in Theorem 3, let s be any integer satisfying $s \geq 8\ln(4T(1+\gamma)m/\delta)/\bar{q}^{*2}_{min}$, and let $\epsilon' = \epsilon^{1-4\sqrt{\ln(4T(1+\gamma)m/\delta)/(2s)}/\bar{q}^*_{min}}$. Then under the perturbation assumption, with probability at least $1 - \delta$, when the Main Algorithm is executed with input $(\epsilon, \gamma, s, \bar{q}^*_{min})$, the following two things hold:*

1) The number of assignments of to non-gold standard tasks is no more than

$$\left(1 + \frac{\min(m,n)}{\hat{q}_{min} n C_{\epsilon'}} + \frac{35\ln(4/\delta)}{\hat{q}_{min}\sqrt{\gamma C_{\epsilon'}}}\right)$$

times the optimal objective of the IP, where $\hat{q}_{min} = \min_{(i,j):y^s_{i,j}=1} \hat{q}^s_{i,j}$.

2) The probability that the aggregated label for each task i is incorrect is bounded by $\epsilon^{(1-l_1)(1-l_2)(1-l_{3,i})}$, where $l_1 = 4t/\bar{q}^*_{min}, l_2 = 6\ln(4/\delta)/\sqrt{\gamma C_{\epsilon'}}$, $l_{3,i} = 4t/\hat{q}_i$, and $t = \sqrt{\ln(4T(1+\gamma)m/\delta)/(2s)}$.

When ϵ is small, $C_{\epsilon'}$ is large, and l_2 approaches 0. The competitive ratio may shrink, but if ϵ is too small, $\hat{q}_{min}$ will shrink as well, and at some point the problem may become infeasible. When s is large, t is small, and so l_1 and $l_{3,i}$ approach 0, leading to error almost as low as if we knew the true $\mathbf{q}$ values, as we would expect.

6. Synthetic Experiments

In this section, we evaluate the performance of our algorithm through simulations on synthetically generated data. As a comparison, we also run the message-passing inference algorithm of Karger et al. (2011a;b) on the same data sets. As described in Section 2, Karger et al. use a non-adaptive, random assignment strategy in conjunction with this inference algorithm. We show that adaptively allocating tasks to workers using our algorithm can outperform random task assignment in settings in which *(i)* the worker distribution is diverse, or *(ii)* the set of tasks is heterogeneous.

We create $n = 1,000$ tasks and $m = 300$ workers with capacity $M_j = 200$ for all j, and vary the distribution over skill levels $p_{i,j}$. We would like to compare the error rates of the algorithms when given access to the same total number of labels. In the message-passing algorithm, we can directly set the number of labels by altering the number of assignments. In our algorithm, we change the parameter ϵ and observe the number of labels (including exploration) and the prediction error.

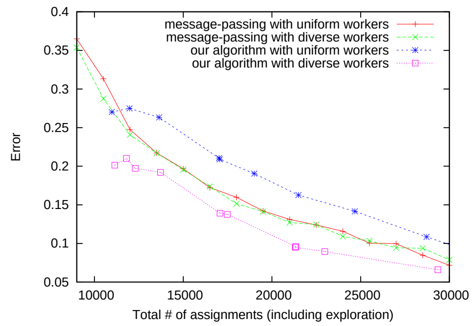

Figure 1. Uniform tasks with one or two worker types.

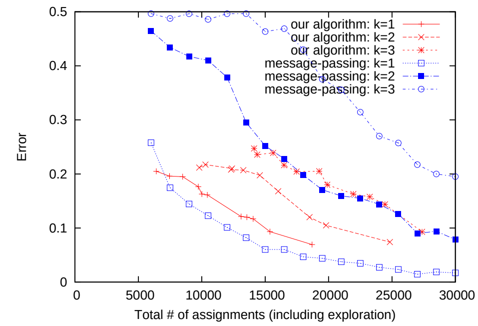

Figure 2. Heterogeneous tasks.

6.1. Worker Diversity

In their analysis, Karger et al. assume there is only one task type (that is, $p_{i,j} = p_{i',j}$ for all i, i', and j), and claim that in this setting adaptively assigning tasks does not yield much of an advantage. Our first experiment simulates this setting. We would like to see if our algorithm can perform better if the worker distribution is diverse, even though it requires some "pure exploration" — we need to pay each worker to complete the gold standard tasks, and we need to hire an extra γm workers to estimate the task weights.

For our algorithm, we set $\gamma = 0.3$ and sample 90 extra workers from the same distribution to learn task weights. Each worker is required to complete $s = 20$ gold standard tasks of each type when she arrives. These values were not optimized, and performance could likely be improved by tuning these parameters.

We examine two settings. In the first, every worker gives us a correct label with probability 0.6414 for all tasks. In the second, the population is 50% spammers and 50% hammers. The spammers give random answers, while the hammers answer correctly with probability 0.7. Note that these values are chosen such that $E[q_{i,j}] = E[(2p_{i,j} - 1)^2]$ is the same in both settings.

The results are shown in Figure 1. The performance of the message-passing algorithm is almost identical in the two settings. Our algorithm performs relatively poorly in the setting with uniform workers since we can't benefit from adaptive assignments but still pay the exploration costs. However, our algorithm outperforms message passing in the setting with two types of workers, quickly learning not to assign any tasks to the spammers beyond those used for exploration.

6.2. Heterogeneous Tasks

We next examine a setting in which there are multiple types of tasks, and every worker is skilled at exactly one type. We generate k task types and k corresponding worker types, for $k = 1$, 2, and 3. Type α workers complete type α tasks correctly with probability 0.7, but other tasks correctly with probability 0.5.

For our algorithm, we set $\gamma = 0.3$. Each worker completes $s = 10$ gold standard tasks of each type.

The results are shown in Figure 2. Not surprisingly, since the message-passing algorithm does not attempt to match tasks to suitable workers, its performance degrades quickly when k grows. Since our algorithm attempts to find the best match between workers and tasks, the performance degrades much more slowly when k grows, even with the extra exploration costs.

7. Conclusion

We conclude by mentioning several extensions of our model. We have assumed that the requester pays the same price for any label. Our results can be extended to handle the case in which different workers charge different prices. Let $c_{i,j}$ denote the cost of obtaining a label for task i from worker j. The objective in the integer program would become $\Sigma_{i=1}^{n} \Sigma_{j=1}^{m} c_{i,j} y_{i,j}$. This is linear and the same techniques would apply.

The framework can also be extended to handle more intricate assumptions about the structure of tasks. We have assumed that there are T task types, with $p_{i,j} = p_{i',j}$ whenever i and i' are of the same type. However, this assumption is used only in the exploration phase in which workers' skills are estimated. While the amount of exploration required by the algorithm depends on the particular task structure assumed, the derivation of our algorithm and the general analysis are independent of the task structure.

Acknowledgements

This research was partially supported by the National Science Foundation under grant IIS-1054911.

References

Alon, Noga, Awerbuch, Baruch, Azar, Yossi, Buchbinder, Niv, and Naor, Joseph. A general approach to online network optimization problems. In *SODA*, 2004.

Bansal, Nikhil, Buchbinder, Niv, and Naor, Joseph. A primal-dual randomized algorithm for weighted paging. In *FOCS*, 2007.

Buchbinder, Niv and Naor, Joseph. Online primal-dual algorithms for covering and packing problems. In *ESA*, 2005.

Buchbinder, Niv, Jain, Kamal, and Naor, Joseph. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA*, 2007.

Crammer, Koby, Kearns, Michael, and Wortman, Jennifer. Learning from data of variable quality. In *NIPS*, 2005.

Crammer, Koby, Kearns, Michael, and Wortman, Jennifer. Learning from multiple sources. *Journal of Machine Learning Research*, 9:1757–1774, 2008.

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

Dekel, Ofer and Shamir, Ohad. Vox populi: Collecting high-quality labels from a crowd. In *COLT*, 2009.

Devanur, Nikhil and Hayes, Thomas. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *ACM EC*, 2009.

Devanur, Nikhil, Jain, Kamal, Sivan, Balasubramanian, and Wilkens, Christopher. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *ACM EC*, 2011.

Ghosh, Arpita, Kale, Satyen, and McAfee, Preston. Who moderates the moderators? Crowdsourcing abuse detection in user-generated content. In *ACM EC*, 2011.

Ho, Chien-Ju and Vaughan, Jennifer Wortman. Online task assignment in crowdsourcing markets. In *AAAI*, 2012.

Ipeirotis, Panagiotis. Analyzing the Amazon Mechanical Turk marketplace. *ACM XRDS*, 17(2):16–21, 2010.

Ipeirotis, Panagiotis G., Provost, Foster, and Wang, Jing. Quality management on Amazon Mechanical Turk. In *HCOMP*, 2010.

Karger, David, Oh, Sewoong, and Shah, Devavrat. Iterative learning for reliable crowdsourcing systems. In *NIPS*, 2011a.

Karger, David, Oh, Sewoong, and Shah, Devavrat. Budget-optimal task allocation for reliable crowdsourcing systems. CoRR, abs/1110.3564, 2011b.

Kittur, Aniket, Chi, Ed, and Suh, Bongwon. Crowdsourcing user studies with Mechanical Turk. In *CHI*, 2008.

Liu, Qiang, Peng, Jian, and Ihler, Alexander. Variational inference for crowdsourcing. In *NIPS*, 2012.

Oleson, Dave, Hester, Vaughn, Sorokin, Alex, Laughlin, Greg, Le, John, and Biewald, Lukas. Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In *HCOMP*, 2011.

Sheng, Victor, Provost, Foster, and Ipeirotis, Panagiotis. Get another label? Improving data quality using multiple, noisy labelers. In *KDD*, 2008.

Tran-Thanh, Long, Stein, Sebastian, Rogers, Alex, and Jennings, Nicholas. Efficient crowdsourcing of unknown experts using multi-armed bandits. In *ICAI*, 2012.

Wah, Catherine, Branson, Steve, Welinder, Peter, Perona, Pietro, and Belongie, Serge. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wais, Paul, Lingamneni, Shivaram, Cook, Duncan, Fennell, Jason, Goldenberg, Benjamin, Lubarov, Daniel, and Martin, David. Towards building a high-quality workforce with mechanical turk. Presented at the *NIPS Workshop on Computational Social Science and the Wisdom of Crowds*, 2010.

Welinder, Peter, Branson, Steve, Belongie, Serge, and Pietro, Perona. The Multidimensional Wisdom of Crowds. In *NIPS*, 2010.

Zhou, Dengyong, Basu, Sumit, Mao, Yi, and Platt, John. Learning from the wisdom of crowds by minimax entropy. In *NIPS*, 2012.

A. Appendix

A.1. Proof of Lemma 1

Without loss of generality, assume that $\ell_i = 1$. Let $X_{i,j}$ be a random variable which represents the weighted label, so

$$X_{i,j} = \begin{cases} w_{i,j}, & \text{with probability } p_{i,j}, \\ -w_{i,j}, & \text{with probability } 1 - p_{i,j}, \end{cases}$$

and let $X_i = \sum_{j \in J_i} X_{i,j}$.

Recall that we predict that task i has label 1 if $X_i \geq 0$ and has label -1 otherwise. Since the true label is 1, bounding $\Pr(X_i \leq 0)$ would give us a bound on the probability of an error. We will need that

$$\begin{aligned} E[X_i] &= \sum_{j \in J_i} E[X_{i,j}] \\ &= \sum_{j \in J_i} (p_{i,j} w_{i,j} - (1 - p_{i,j}) w_{i,j}) \\ &= \sum_{j \in J_i} (w_{i,j}(2p_{i,j} - 1)). \end{aligned}$$

Note that $\Pr(X_i \leq 0) = \Pr(E[X_i] - X_i \geq E[X_i])$. By Hoeffding's inequality, we have

$$\begin{aligned} \Pr(X_i \leq 0) &\leq \exp\left(-\frac{2(E[X_i])^2}{\sum_{j \in J_i} (2w_{i,j})^2} \right) \\ &= \exp\left(-\frac{(\sum_{j \in J_i} w_{i,j}(2p_{i,j} - 1))^2}{2 \sum_{j \in J_i} w_{i,j}^2} \right). \end{aligned}$$

This shows the first part of this lemma. This error bound is maximized when the expression

$$\frac{(\sum_{j \in J_i} w_{i,j}(2p_{i,j} - 1))^2}{2 \sum_{j \in J_i} w_{i,j}^2}$$

is minimized. Setting the gradient of this expression to $\mathbf{0}$, we see that this happens when for every k,

$$\frac{\sum_{j \in J_i} w_{i,j}(2p_{i,j} - 1)}{\sum_{j \in J_i} w_{i,j}^2} w_{i,k} = 2p_{i,k} - 1,$$

i.e., when $w_{i,k} \propto 2p_{i,k} - 1$. (Note that scaling the weights will not change the bound since $\hat{\ell}_i$ will not change.) Plugging $w_{i,j} = 2p_{i,j} - 1$ into the bound from the first half of the lemma, we get that

$$\begin{aligned} \Pr(X_i \leq 0) &\leq \exp\left(-\frac{(\sum_{j \in J_i} q_{i,j})^2}{2 \sum_{j \in J_i} q_{i,j}} \right) \\ &= \exp\left(-\frac{1}{2} \sum_{j \in J_i} q_{i,j} \right). \end{aligned}$$

□

A.2. Proof of Theorem 1

The bulk of this proof involves showing that there exists a primal optimal $\bar{\mathbf{y}}^*$ for the relaxed linear program such that for at most $\min(m, n)$ pairs of (i, j), $\bar{y}^*_{i,j} \neq y_{i,j}$. Therefore, since $\bar{y}^*_{i,j}, y_{i,j} \in [0, 1]$ for all i and j, $\sum_{i=1}^n \sum_{j=1}^m (y_{i,j} - \bar{y}^*_{i,j}) \leq \min(m, n)$. To complete the proof, we will use the fact that $\mathbf{y}^*$ is feasible in the relaxed linear program to show that this implies the result.

We start with a helpful lemma that characterizes the dual optimal solution of the relaxed linear program.

Lemma 4. *If $\boldsymbol{x}^*$ is the optimal value of $\boldsymbol{x}$ in any dual optimal solution, then there exists a dual optimal solution $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{t}^*)$ such that for each $j \in \{1, \cdots, m\}$,*

$$z_j^* = \begin{cases} 0, & \text{if } n_j \leq M_j, \\ v_{c_j, j}, & \text{otherwise} \end{cases}$$

where $v_{i,j} = q_{i,j} x_i^* - 1$ is the task value for (i, j) and c_j is the task with the M_jth largest task value for worker j among all tasks.

Proof. We prove the lemma in two steps. In Step 1, we show that there cannot exist a dual optimal $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ such that $\bar{z}_j \neq v_{c_j, j}$ for some j such that $n_j > M_j$. In Step 2, we show that if there exists a dual optimal $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ such that $\bar{z}_j \neq 0$ for some j with $n_j \leq M_j$, then there exists another dual optimal solution $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ such that

$$\bar{z}'_k = \begin{cases} \bar{z}_k, & \text{if } k \neq j, \\ 0, & \text{if } k = j, \end{cases} \tag{4}$$

and

$$\bar{t}'_{i,k} = \begin{cases} \bar{t}_{i,k}, & \text{if } k \neq j \text{ or } k = j \text{ and } v_{i,j} < 0, \\ \bar{t}_{i,k} + \bar{z}_j, & \text{otherwise.} \end{cases} \tag{5}$$

Therefore, starting with any dual solution, we can transform it into a solution satisfying the condition that $z_j^* = 0$ if $n_j \leq M_j$ by repeating this transformation for any workers j for which the condition did not originally hold.

In the following proof, we define the function g to represent the dual objective:

$$g(\mathbf{x}, \mathbf{z}, \mathbf{t}) = C_\epsilon \sum_{i=1}^n x_i - \sum_{j=1}^m M_j z_j - \sum_{i=1}^n \sum_{j=1}^m t_{i,j}.$$

Recall that the dual constraints are

$$\begin{aligned} &1 - q_{i,j} x_i + z_j + t_{i,j} \geq 0 \ \forall (i, j), \\ &x_i, z_j, t_{i,j} \geq 0. \end{aligned}$$

Step 1: Assume by contradiction that there exists a dual optimal $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ and worker j such that $\bar{z}_j \neq v_{c_j,j}$ and $n_j > M_j$. Below we show that we can always generate a dual feasible solution $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ which leads to higher dual objective. Therefore, $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ cannot be optimal.

Define

$$\bar{z}'_k = \begin{cases} \bar{z}_k, & \text{if } k \neq j, \\ v_{c_j,j}, & \text{if } k = j, \end{cases}$$

and

$$\bar{t}'_{i,k} = \begin{cases} \bar{t}_{i,k}, & \text{if } k \neq j \text{ or } k = j \text{ and } v_{i,j} < 0 \\ & \text{or } k = j \text{ and } v_{i,j} \geq 0 \text{ and } \bar{z}_j > v_{c_j,j}, \\ \bar{t}_{i,k} + v_{c_j,j} - \bar{z}_j, & \text{otherwise.} \end{cases}$$

1) Suppose that $\bar{z}_j < v_{c_j,j}$. We can observe that $\bar{z}'_j > \bar{z}_j$ and $\bar{t}'_{i,k} \geq \bar{t}_{i,k}$ for all k. Since $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ is dual feasible, We can verify that $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ is also dual feasible.

Next we show that $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ leads to higher dual objective than $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$.

$$\begin{aligned} & g(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}}) - g(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}') \\ &= -M_j \bar{z}_j + M_j v_{c_j,j} - \sum_{i:v_{i,j} \geq 0} v_{c_j,j} + \sum_{i:v_{i,j} \geq 0} \bar{z}_j \\ &= (\bar{z}_j - v_{c_j,j})(n_j - M_j) < 0. \end{aligned}$$

Therefore, $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ couldn't have been optimal.

2) Suppose that $\bar{z}_j > v_{c_j,j}$. Note that $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ is dual feasible, and the only change in $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ is that $\bar{z}'_j = v_{c_j,j}$. We need only examine if $-v_{c_j,j} + \bar{z}'_j + \bar{t}_{i,j} \geq 0$ to check the dual feasibility. Since $\bar{t}'_{i,j} \geq 0$, we know $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ is dual feasible.

Furthermore, $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ also leads to higher dual objective:

$$\begin{aligned} g(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}}) - g(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}') &= -M_j \bar{z}_j + M_j v_{c_j,j} \\ &= M_j (v_{c_j,j} - \bar{z}_j) < 0. \end{aligned}$$

Therefore, $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ couldn't have been optimal. This shows that there cannot be a dual optimal solution with $z^*_j \neq v_{c_j,j}$ for some j with $n_j > M_j$.

Step 2: Assume there exists a dual optimal solution $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ and worker j such that $\bar{z}_j > 0$ and $n_j \leq M_j$. Below we show that $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$, as defined in Equations 4 and 5, is dual feasible, and the dual objective $g(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ is no less than $g(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$.

To show feasibility, first observe that for all $k \neq j$, the dual variables $\bar{z}'_k$ and $\bar{t}'_{i,k}$ are not changed. Therefore, we need only check if the constraint $-v_{i,k} + \bar{z}'_k + \bar{t}'_{i,k} \geq 0$ holds for all i and $k = j$. Observing Equations 4 and 5, when $v_{i,j} < 0$, the constraint trivially holds since $\bar{z}'_k$ and $\bar{t}'_{i,k}$ are nonnegative. When $v_{i,j} \geq 0$, the left-hand side of the constraint can be written as $v_{i,j} + \bar{t}_{i,k} + \bar{z}_j$, which is larger then or equal to 0 since $(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}})$ is dual feasible. Therefore, $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ is also dual feasible.

Furthermore, we have

$$\begin{aligned} g(\mathbf{x}^*, \bar{\mathbf{z}}, \bar{\mathbf{t}}) - g(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}') &= -M_j \bar{z}_j + \sum_{i:v_{i,j} \geq 0} \bar{z}_j \\ &= (-M_j + n_j)\bar{z}_j \leq 0. \end{aligned}$$

Therefore, $(\mathbf{x}^*, \bar{\mathbf{z}}', \bar{\mathbf{t}}')$ must be an optimal dual solution.

Combining this with Step 1, we can always transform any dual solution $(\mathbf{x}^*, \mathbf{z}^*, \mathbf{t}^*)$ into another that satisfies the properties in the lemma without changing $\mathbf{x}^*$. □

Given the dual optimal above, using complementary slackness, we can characterizes the primal optimal solution $\bar{\mathbf{y}}^*$. Below we list the cases in which $\bar{y}^*_{i,j}$ takes on integer values. Looking ahead, we will see that in all of these cases, $\bar{y}^*_{i,j}$ and $y_{i,j}$ do not differ. We can then show there are at most $\min(m, n)$ pairs of $y_{i,j}$ such that $y_{i,j} \neq \bar{y}^*_{i,j}$.

Lemma 5. *Consider a dual optimal solution $(\boldsymbol{x}^*, \boldsymbol{z}^*, \boldsymbol{t}^*)$ satisfying the criteria in Lemma 4, and let $\bar{\boldsymbol{y}}^*$ be the corresponding primal optimal solution to the relaxed linear programming problem. Let $v_{i,j} = q_{i,j} x^*_i - 1$ be the task value for (i, j) and c_j be the task with the M_jth largest task value for worker j among all tasks. For all (i, j),*

1. *If $v_{i,j} < 0$ then $\bar{y}^*_{i,j} = 0$.*
2. *If $v_{i,j} > 0$ & $n_j \leq M_j$ then $\bar{y}^*_{i,j} = 1$.*
3. *If $v_{i,j} > 0$, $n_j > M_j$, & $v_{i,j} > v_{c_j,j}$, then $\bar{y}^*_{i,j} = 1$.*
4. *If $v_{i,j} > 0$, $n_j > M_j$, & $v_{i,j} < v_{c_j,j}$, then $\bar{y}^*_{i,j} = 0$.*

Proof. We prove the four parts in turn. First, it is useful to note that by complementary slackness, we have

$$\bar{y}^*_{i,j}(1 - q_{i,j} x^*_i + z^*_j + t^*_{i,j}) = 0 \;\; \forall (i, j), \tag{6}$$
$$t^*_{i,j}(\bar{y}^*_{i,j} - 1) = 0 \;\; \forall (i, j). \tag{7}$$

Part 1: The first part is simple. Since $z^*_j, t^*_{ij} \geq 0$, if $v_{i,j} < 0$, it must be the case that $1 - q_{i,j} x^*_i + z^*_j + t^*_{i,j} > 0$. From Equation 6, we have $\bar{y}^*_{i,j} = 0$.

Part 2: From Lemma 4, if $n_j \leq M_j$ then $z^*_j = 0$. When $z^*_j = 0$ and $v_{i,j} > 0$, we must have that $t^*_{i,j} \geq v_{i,j}$ in order to satisfy the first dual constraint. Equation 7 then implies that $\bar{y}^*_{i,j} = 1$.

Part 3: From Lemma 4, if $n_j > M_j$ then $z^*_j = v_{c_j,j}$. In this case, when $v_{i,j} > v_{c_j,j}$, by the dual constraints we must have $t^*_{i,j} \geq v_{i,j} - v_{c_j,j} > 0$, and by Equation 7, $\bar{y}^*_{i,j} = 1$.

Part 4: Again, from Lemma 4, since $n_j > M_j$ we know that $z^*_j = v_{c_j,j}$. In this case, if $v_{i,j} < v_{i',j}$, then any $t_{i,j} \geq 0$ is feasible. Since $t_{i,j}$ appears negated in the objective and does not appear in other constraints, we must have $t^*_{i,j} = 0$. This implies that $1 - q_{i,j}x^*_i + z^*_j + t^*_{i,j} > 0$, and by Equation 6, $\bar{y}^*_{i,j} = 0$. □

One can verify that in all of the cases covered by Lemma 5, $\bar{y}^*_{i,j} = y_{i,j}$. Notice that the only case not covered by this lemma is the case in which $v_{i,j} > 0$, $n_j > M_j$, and $v_{i,j} = v_{c_j,j}$. Since we have assumed that $v_{i,j} \neq v_{i',j}$ unless $i = i'$, this can happen at most once for each worker j, and, therefore, at most m times in total. In cases where $y_{i,j}$ and $\bar{y}^*_{i,j}$ differ, $y_{i,j} = 1$ and $\bar{y}^*_{i,j} \in [0, 1]$. Hence,

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{m} \bar{y}^*_{i,j} \leq m. \tag{8}$$

Since $y_{i,j} \in \{0, 1\}$, we know that for each worker j, there exists at most one task i such that $\bar{y}^*_{i,j} \notin \{0, 1\}$. Furthermore, we know that if such a task exists, the Primal Approximation Algorithm assigns the task to worker j, i.e., $y_{i,j} = 1$. Below we further argue that for each task i, there exists at most one worker j such that $\bar{y}^*_{i,j} \notin \{0, 1\}$, and this worker is the one with minimum skill level $q_{i,j}$ among all the workers with $y_{i,j} = 1$. Hence, the difference in Equation 8 can be upper bounded by n as well.

Let j_i be the worker with the minimum skill level among all the workers with $y_{i,j} = 1$. As we will describe in Section A.3, we can add random noise to the $q_{i,j}$ values such that $q_{i,j} \neq q_{i',j'}$ unless $i = i'$ and $j = j'$. Assume for contradiction that there exists a worker $j \neq j_i$ such that $\bar{y}^*_{i,j} \notin \{0, 1\}$. We know that the Primal Approximation Algorithm sets $y_{i,j}$ to 1 when $\bar{y}^*_{i,j} \notin \{0, 1\}$. Therefore, we have $q_{i,j} > q_{i,j_i}$ by definition of j_i. Let $\Delta = \min(q_{i,j_i}\bar{y}^*_{i,j_i}/q_{i,j}, 1 - \bar{y}^*_{i,j})$. Suppose that we increased the value of $\bar{y}^*_{i,j}$ by Δ and decreased the value of $\bar{y}^*_{i,j_i}$ by $q_{i,j}\Delta/q_{i,j_i}$. It is easy to verify that no constraints of the relaxed offline formulation would be violated, but the objective would decrease since $q_{i,j} > q_{i,j_i}$. This is a contradiction. Therefore, for each task i, there exists at most one worker j such that $\bar{y}^*_{i,j} \notin \{0, 1\}$, and this worker is the one with minimum skill level $q_{i,j}$ among all the workers with $y_{i,j} = 1$. Hence,

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{m} \bar{y}^*_{i,j} \leq n. \tag{9}$$

Combining Equations 8 and 9

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{m} \bar{y}^*_{i,j} \leq \min(m, n).$$

Finally, since $\mathbf{y}^*$ (the optimal solution of the IP) is feasible in the relaxed linear program where $\bar{\mathbf{y}}^*$ is optimal, we know that

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \bar{y}^*_{i,j} \leq \sum_{i=1}^{n}\sum_{j=1}^{m} y^*_{i,j}.$$

Putting these together, we have

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{i,j} - \sum_{i=1}^{n}\sum_{j=1}^{m} y^*_{i,j} \leq \min(m, n),$$

which completes the proof of Theorem 1. □

A.3. Discussion of Perturbation Assumption

In Section 4.3 where we introduced the perturbation assumption, we claimed that we can add small random perturbations (noise) to the $q_{i,j}$ values as in Devanur & Hayes (2009) to satisfy the condition that $q_{i,j}x_i \neq q_{i',j}x_{i'}$ for $i \neq i'$. In this section, we explain the details of the claim.

First of all, note that if all the $q_{i,j}$ values *(i)* are determined independently, and, *(ii)* are in general position in $\mathbb{R}$, then the probability that any two $q_{i,j}$ values are the same is 0 (Devanur & Hayes, 2009). However, in our solution, the $q_{i,j}$ values are estimated by our exploration phase. Since the number of exploratory assignments are small, there is only a limited number of possible $q_{i,j}$ values. Therefore, the probability of the $q_{i,j}$ values being all different is clearly *non-zero*. To remedy this, we can add a small amount of noise to the $q_{i,j}$ values after estimating them empirically. If the noise is drawn from a uniform distribution, the new noisy $q_{i,j}$ values will be in general position. Therefore, the probability that two noisy $q_{i,j}$ values are the same is 0 after adding the noise. Note that we can set the random noise arbitrarily small to reduce the effects of it in our analysis. For example, if we know the error in estimating the $q_{i,j}$ values is bounded by t, then the amount of random noise can be set to something much smaller than t.

Second, we show that when $q_{i,j}$ values are all different, at most $n - 1$ ties in the $q_{i,j}x^*_i$ values can happen.

Note that for some worker j and two tasks i and i', if $q_{i,j}x_i^* = q_{i',j}x_{i'}^*$, then $x_i^* = x_{i'}^*(q_{i',j}/q_{i,j})$. Hence, in the worst case, an adversary could create at most $n-1$ ties by adjusting the $\mathbf{x}^*$ values since the noisy $q_{i,j}$ values are in general position.

A.4. Proof of Theorem 2

We first characterize some properties of the Primal Approximation Algorithm. In order to do this, we consider the expanded version of the algorithm in Algorithm 3. Its behavior is identical to the algorithm stated in Section 4.3 in terms of the way that the assignments $y_{i,j}$ are made, but it also includes settings for the dual variables $\mathbf{z}$ and $\mathbf{t}$ as well.

Algorithm 3 The Expanded Primal Approximation Algorithm, which explicitly sets dual variables.

```
Input: Values x_i and q_{i,j} for all (i, j)
Output: Values y_{i,j}, t_{i,j}, z_j for all (i, j)
Calculate task values v_{i,j} = q_{i,j} x_i − 1 for all i.
if there are no more than M_j tasks with v_{i,j} ≥ 0 then
  Aet z_j to 0.
  for every task i with v_{i,j} ≥ 0 do
    Set t_{i,j} = v_{i,j} = q_{i,j} x_i − 1.
    Set y_{i,j} = 1.
  end for
  for every task i with v_{i,j} < 0 do
    Set t_{i,j} = 0.
    Set y_{i,j} = 0.
  end for
end if
if there are more than M_j tasks with v_{i,j} ≥ 0 then
  Set z_j to a value such that there are exactly M_j
  tasks with v_{i,j} − z_j ≥ 0.
  for every task i with v_{i,j} − z_j ≥ 0 do
    Set t_{i,j} = v_{i,j} − z_j = q_{i,j} x_i − 1 − z_j.
    Set y_{i,j} = 1.
  end for
  for every task i with v_{i,j} − z_j < 0 do
    Set t_{i,j} = 0.
    Set y_{i,j} = 0.
  end for
end if
```

This expanded version of the algorithm can be used to prove the following lemma.

Lemma 6. *Given any inputs $\boldsymbol{x}$ and $\boldsymbol{q}$, let $\boldsymbol{y}$ and $(\boldsymbol{z}, \boldsymbol{t})$ be the primal assignments and dual variables set by the Expanded Primal Approximation Algorithm. Let $P(\boldsymbol{y})$ and $D(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{t})$ denote the primal and dual objective. Then under the perturbation assumption,*

$$P(\boldsymbol{y}) = D(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{t}) + \sum_{i=1}^{n}\left(\sum_{j=1}^{m} q_{i,j}y_{i,j} - C_\epsilon\right)x_i.$$

Proof. By examining the Expanded Primal Approximation Algorithm, one can easily verify the following properties.

1. If $t_{i,j} > 0$, then $y_{i,j} = 1$.
2. $\sum_{i=1}^n y_{i,j} \le M_j$.
3. If $\sum_{i=1}^n y_{i,j} < M_j$, then $z_j = 0$.
4. If $y_{i,j} = 1$, then $z_j + t_{i,j} = q_{i,j}x_i - 1$.

By definition, the dual objective can be written as

$$D(\mathbf{x}, \mathbf{z}, \mathbf{t}) = C_\epsilon \sum_{i=1}^n x_i - \sum_{j=1}^m M_j z_j - \sum_{i=1}^n \sum_{j=1}^m t_{i,j}.$$

Note that the first property implies $\Sigma_{i=1}^n \Sigma_{j=1}^m t_{i,j} = \Sigma_{i=1}^n \Sigma_{j=1}^m t_{i,j} y_{i,j}$. The second and third properties together imply $M_j z_j = (\Sigma_{i=1}^n y_{i,j}) z_j$. Hence,

$$\begin{aligned} D(\mathbf{x}, \mathbf{z}, \mathbf{t}) &= C_\epsilon \sum_{i=1}^n x_i - \sum_{j=1}^m (\sum_{i=1}^n y_{i,j}) z_j - \sum_{i=1}^n \sum_{j=1}^m t_{i,j} y_{i,j} \\ &= C_\epsilon \sum_{i=1}^n x_i - \sum_{i=1}^n \sum_{j=1}^m (z_j + t_{i,j}) y_{i,j}. \end{aligned}$$

Finally, since $y_{i,j} \in \{0, 1\}$, the fourth property implies that $(z_j + t_{i,j}) y_{i,j} = (q_{i,j} x_i - 1) y_{i,j}$. Therefore,

$$\begin{aligned} D(\mathbf{x}, \mathbf{z}, \mathbf{t}) &= C_\epsilon \sum_{i=1}^n x_i - \sum_{i=1}^n \sum_{j=1}^m (q_{i,j} x_i - 1) y_{i,j} \\ &= C_\epsilon \sum_{i=1}^n x_i - \sum_{i=1}^n \sum_{j=1}^m (q_{i,j} y_{i,j}) x_i + \sum_{i=1}^n \sum_{j=1}^m y_{i,j} \\ &= P(\mathbf{y}) - \sum_{i=1}^n \left(\sum_{j=1}^m q_{i,j} y_{i,j} - C_\epsilon\right) x_i. \end{aligned}$$

□

In our algorithm, we hire an extra γm workers and observe their skill levels $q_{i,j}$. We then solve the sampled LP, which consists of these γm workers. Let $\hat{\mathbf{x}}^*$ be the optimal task weights of the sampled LP. We then apply the Primal Approximation Algorithm with inputs $\hat{\mathbf{x}}^*$ and $q_{i,j}$ values of the primary m workers. We denote $\hat{\mathbf{y}}^*$ as the assignments generated by the Primal Approximation Algorithm using these two inputs.

By observing Algorithm 3, we can see that as long as the input x_i is non-negative for all i, z_j and $t_{i,j}$ will be non-negative and $1 - q_{i,j}x_i + z_j + t_{i,j} \geq 0$. Therefore, the Primal Approximation Algorithm always generates feasible dual solutions, and the dual objective $D(\mathbf{x}, \mathbf{z}, \mathbf{t})$ is no bigger than primal optimal by weak duality for any inputs $\mathbf{x}$ and $\mathbf{q}$, subject to $x_i \geq 0$ for all i.

Next we bound the values of $\hat{x}_i^*$ (in Lemma 7) and $\Sigma_{j=1}^m q_{i,j}\hat{y}_{i,j}^* - C_\epsilon$ (in Lemma 9). And then we can show the primal objective is close to optimal.

Lemma 7. *For every $i \in \{1, \cdots, n\}$ and $j \in \{1, \cdots, \gamma m\}$, let $q_{i,j}^s$ be a value in $[0,1]$ perturbed by random noise so that $q_{i,j}^s \neq q_{i',j'}^s$ unless $i = i'$ and $j = j'$. Let $\hat{\boldsymbol{x}}^*$ be the optimal task weights of the sampled LP with these perturbed values. Let $\hat{\boldsymbol{y}}^{s,*}$ denote the assignment generated by applying the Primal Approximation Algorithm with inputs $\hat{\boldsymbol{x}}^*$ and $\boldsymbol{q}^s$. Then under the perturbation assumption, for all i, $\hat{x}_i^* = 1/\min_{\{j:\hat{y}_{i,j}^{s,*}=1\}} q_{i,j}^s$.*

Proof. In this proof, our discussion is restricted to the sampled LP. Therefore, we omit all the superscript s in the notations.

Let j_i be the worker with minimum skill level for task i among the set of workers assigned to task i, i.e., $j_i = \text{argmin}_{j:\hat{y}_{i,j}^*=1} q_{i,j}$. Note that j_i is uniquely specified since all $q_{i,j}$ values are different.

Let $\bar{\mathbf{y}}^*$ be the optimal solution of the sampled LP. By the same argument as the one in the proof of Theorem 1, we know that for each worker j, there exists at most one task i such that $\bar{y}_{i,j}^* \neq \hat{y}_{i,j}^*$, which implies that for each worker j, there exists at most one task i such that $\bar{y}_{i,j}^* \notin \{0,1\}$. Furthermore, we know that if such a task exists, the Primal Assignment Algorithm assigns the task to worker j, i.e., $\hat{y}_{i,j}^* = 1$.

We now show that if $\bar{y}_{i,j}^* \notin \{0,1\}$, then $j = j_i$. This means for each task i, there exists at most one worker j such that $\bar{y}_{i,j}^* \notin \{0,1\}$, and this worker is the one with minimum skill level among all the workers who are assigned to that task. Assume for contradiction that there exists a worker $j \neq j_i$ such that $\bar{y}_{i,j}^* \in (0,1)$. We know that this worker is assigned to task i by the Primal Approximation Algorithm and so $q_{i,j} > q_{i,j_i}$. Let $\Delta = \min(q_{i,j_i}\bar{y}_{i,j_i}^*/q_{i,j}, 1 - \bar{y}_{i,j}^*)$. Suppose that we increased the value of $\bar{y}_{i,j}^*$ by Δ and decreased the value of $\bar{y}_{i,j_i}^*$ by $q_{i,j}\Delta/q_{i,j_i}$. It is easy to verify that no constraints of the sampled LP would be violated, but the objective would increase since $q_{i,j} > q_{i,j_i}$. This is a contradiction.

Since the $q_{i,j}$ values have been randomly perturbed, there cannot exist any set of workers J_i for any task i such that $\Sigma_{j\in J_i} q_{i,j} = C_\epsilon$. Together with the last paragraph and the fact that $\Sigma_j q_{i,j}\bar{y}_{i,j}^* = C_\epsilon$, this implies that for each task i, $\bar{y}_{i,j_i}^* \notin \{0,1\}$.

With this fact in place, we are ready to show that $\hat{x}_i^* = 1/q_{i,j_i}$ for all i. Let $(\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*, \hat{\mathbf{t}}^*)$ denote the dual optimal solution of the sampled LP.

First, assume there exists a task i such that $\hat{x}_i^* > 1/q_{i,j_i}$. Then we know that $1 - q_{i,j_i}\hat{x}_i^* < 0$. Recall that by the dual constraints, $1 - q_{i,j}\hat{x}_i^* + \hat{z}_j^* + \hat{t}_{i,j}^* \geq 0$ for all (i,j). Therefore, we have either $\hat{z}_{j_i}^* > 0$ or $\hat{t}_{i,j_i}^* > 0$. However, by complementary slackness, if $\hat{t}_{i,j_i}^* > 0$, we have $\bar{y}_{i,j_i}^* = 1$, which is a contradiction to the property above. On the other hand, if $\hat{z}_{j_i}^* > 0$, then $\Sigma_{i=1}^n \bar{y}_{i,j_i}^* = M_j$. Since M_j is an integer and there exist at most one task i such that $\bar{y}_{i,j_i}^* \notin \{0,1\}$, we have $\bar{y}_{i,j_i}^* \in \{0,1\}$. This is also a contradiction. Therefore, we must have that $\hat{x}_i^* \leq 1/q_{i,j_i}$.

Now assume there exists a task i such that $\hat{x}_i^* < 1/q_{i,j_i}$. Then $1 - q_{i,j_i}\hat{x}_i^* > 0$. Therefore $1 - q_{i,j_i}\hat{x}_i^* + \hat{z}_{j_i}^* + \hat{t}_{i,j_i}^* > 0$, which, by complementary slackness, means $\bar{y}_{i,j_i}^* = 0$. This is again a contradiction.

From the previous two paragraphs, we know that $\hat{x}_i^* = 1/q_{i,j_i}$ for all i. □

Next, we bound the value of $\Sigma_{j=1}^m q_{i,j}\hat{y}_{i,j}^* - C_\epsilon$. The proof relies on Lemma 3 of (Devanur & Hayes, 2009), which we restate here for completeness.

Lemma 8. *(Devanur & Hayes, 2009) Let $Y = (Y_1, ..., Y_m)$ be a vector of random variables. For any $\gamma \in (0,1)$ such that γm is an integer, let S be a random subset of $\{1, \ldots, m\}$ of size γm, and $Y_S = \sum_{j\in S} Y_j$. Then under the perturbation assumption, for any $\delta \in (0,1)$,*

$$Pr(|Y_S - E[Y_S]| \geq \frac{2}{3}\|Y\|_\infty \ln\frac{2}{\delta} + \|Y\|_2\sqrt{2\gamma\ln\frac{2}{\delta}}) \leq \delta.$$

Lemma 9. *For any given task i, with probability of at least $1-\delta$,*

$$\sum_{j=1}^m q_{i,j}\hat{y}_{i,j}^* - C_\epsilon \leq \frac{d_{m_i}}{\gamma} + \frac{20}{\gamma}\ln\frac{2}{\delta} + 10\sqrt{\frac{1}{\gamma}\ln\frac{2}{\delta}}\sqrt{C_\epsilon + \frac{1}{\gamma}}.$$

where $\Sigma_{i=1}^n d_{m_i} = \gamma m$. Since $\gamma \geq 1/C_\epsilon$,

$$\sum_{j=1}^m q_{i,j}\hat{y}_{i,j}^* - C_\epsilon \leq \frac{d_{m_i}}{\gamma} + 35\ln\frac{2}{\delta}\sqrt{\frac{C_\epsilon}{\gamma}}.$$

Proof. We are considering the random permutation model. The adversary chooses values of capacities M_j and skills $p_{i,j}$ for $(1+\gamma)m$ workers, but the workers' arriving sequence is randomly permuted. In our algorithm, we hire the first γm workers and estimate the task weights. We then try to minimize the number of assignments for the latter m workers.

To simplify notations, in the proof, we use $\mathbf{q}^s$ to denote the skills of the first γm workers, and use $\mathbf{q}$ to denote the skill levels of primary m workers. We will use the following two facts to prove the lemma.

1. If $\hat{\mathbf{x}}^*$ is the optimal task weights of the sampled LP, let $\hat{\mathbf{y}}^{s,*}$ be the assignments generated by the Primal Approximation Algorithm with input $\hat{\mathbf{x}}^*$ and $\mathbf{q}^s$. We have $\gamma C_\epsilon \le \Sigma_{j=1}^{\gamma m} q_{i,j}^s \hat{y}_{i,j}^{s,*} \le \gamma C_\epsilon + 1$. This property follows directly from the proof of Lemma 7. For each task i, there is at most one value $\hat{y}_{i,j}^{s,*}$ which is not equal to the optimal value $y_{i,j}^{s,*}$ of the sampled LP, and $\Sigma_{j=1}^{\gamma m} q_{i,j}^s y_{i,j}^{s,*} = C_\epsilon$.

   We also have $\Sigma_{j=1}^{\gamma m} q_{i,j}^s \hat{y}_{i,j}^{s,*} \le \gamma C_\epsilon + d_{m_i}$, where $\Sigma_{i=1}^{n} d_{m_i} = \gamma m$. This property follows directly from the proof of Theorem 1. For worker j, there is at most one $\hat{y}_{i,j}^{s,*}$ which is not equal to the optimal solution of the sampled LP.

2. Given any task weight $\mathbf{x}$, let $\mathbf{y}^s$ be the assignments generated by the Primal Approximation Algorithm with inputs $\mathbf{x}$ and $\mathbf{q}^s$, and $\mathbf{y}$ be the assignments generated by the Primal Approximation Algorithm with input $\mathbf{x}$ and $\mathbf{q}$. Note that in our setting, given task weights $\mathbf{x}$ and worker skill levels, the assignments $y_{i,j}$ made by the Primal Approximation Algorithm are determined and remain the same for a given task, whether it appears in the initial sample or primary set of workers. Below we use this fact to show that, for task i, with high probability, $\Sigma_{j=1}^{\gamma m} q_{i,j}^s y_{i,j}^s$ is close to $\gamma \Sigma_{j=1}^{m} q_{i,j} y_{i,j}$.

   Consider a single task i. Define $Y_j = q_{i,j}^s y_{i,j}^s$ for $j \in \{1, ..., \gamma m\}$ and $Y_j = q_{i,j-\gamma m} y_{i,j-\gamma m}$ for $j \in \{\gamma m + 1, ..., (1+\gamma)m\}$, and let $Y = \{Y_1, ... Y_{(1+\gamma)m}\}$. Since we assume the worker arriving sequence is randomly permuted, we can think of $\Sigma_{j=1}^{\gamma m} q_{i,j}^s y_{i,j}^s$ as the sum of a random subset of size γm from Y. We can then obtain a bound by applying Lemma 8.

   Before applying the lemma, we first bound the values of $\|Y\|_\infty$, $\|Y\|_2$, and $E[Y_S]$. Below we use the notation $Q_i = \Sigma_{j=1}^{m} q_{i,j} y_{i,j}$ and $Q_i^s = \Sigma_{j=1}^{\gamma m} q_{i,j}^s y_{i,j}^s$.

$$\|Y\|_\infty = \max_j \{Y_j\} \le 1.$$

$$\begin{aligned}\|Y\|_2 &= \sqrt{\sum_{j=1}^{m} (q_{i,j} y_{i,j})^2 + \sum_{j=1}^{\gamma m} (q_{i,j}^s y_{i,j}^s)^2} \\ &\le \sqrt{Q_i + Q_i^s}.\end{aligned}$$

$$E[Y_S] = \frac{\gamma}{(1+\gamma)} (Q_i + Q_i^s).$$

   Applying Lemma 8, with probability at least $1-\delta$,

$$\begin{aligned}&|Q_i^s - \frac{\gamma}{1+\gamma}(Q_i + Q_i^s)| \\ \le& \frac{2}{3} \ln \frac{2}{\delta} + \sqrt{Q_i + Q_i^s} \sqrt{2 \frac{\gamma}{1+\gamma} \ln \frac{2}{\delta}}.\end{aligned}$$

   Finally, multiplying both sides by $(1+\gamma)$ gives us

$$\begin{aligned}&|\gamma Q_i - Q_i^s| \\ \le& (1+\gamma) \frac{2}{3} \ln \frac{2}{\delta} + \sqrt{Q_i + Q_i^s} \sqrt{2\gamma(1+\gamma) \ln \frac{2}{\delta}}.\end{aligned}$$

Let $Q_i^* = \Sigma_{j=1}^{m} q_{i,j} \hat{y}_{i,j}^*$ and $Q_i^{s,*} = \Sigma_{j=1}^{\gamma m} q_{i,j}^s \hat{y}_{i,j}^{s,*}$. Using the two facts above, we have

$$Q_i^* \le \frac{Q_i^{s,*}}{\gamma} + \frac{1+\gamma}{\gamma} \frac{2}{3} \ln \frac{2}{\delta} + \sqrt{\frac{2(1+\gamma)}{\gamma} \ln \frac{2}{\delta}} \sqrt{Q_i^* + Q_i^{s,*}}.$$

To simplify the notation, let $C = Q_i^{s,*}/\gamma$ and $K = 2\ln(2/\delta)(1+\gamma)/\gamma$, we can get

$$\begin{aligned}Q_i^* &\le C + K/3 + \sqrt{K}\sqrt{\gamma C + Q_i^*} \\ &\le C + K/3 + \sqrt{K}\sqrt{\gamma C} + \sqrt{K}\sqrt{Q_i^*}.\end{aligned}$$

By rearranging the variables, we can get

$$Q_i^* - \sqrt{K}\sqrt{Q_i^*} - (C + K/3 + \sqrt{K}\sqrt{\gamma C}) \le 0.$$

Applying the quadratic equation, we know that

$$\begin{aligned}&\frac{\sqrt{K} - \sqrt{K + 4(C + K/3 + \sqrt{K}\sqrt{\gamma C})}}{2} \\ \le& \sqrt{Q_i^*} \le \frac{\sqrt{K} + \sqrt{K + 4(C + K/3 + \sqrt{K}\sqrt{\gamma C})}}{2}.\end{aligned}$$

Therefore,

$$\begin{aligned}\sqrt{Q_i^*} &\le \frac{\sqrt{K} + \sqrt{K} + 2\sqrt{C + K/3 + \sqrt{K}\sqrt{\gamma C}}}{2} \\ &= \sqrt{K} + \sqrt{C + K/3 + \sqrt{K}\sqrt{\gamma C}}.\end{aligned}$$

Since $\gamma \le 1$, taking the square on both sides,

$$Q_i^* \le K + C + K/3 + \sqrt{K\gamma C} + 2\sqrt{K}\sqrt{C + K/3 + \sqrt{K\gamma C}}$$
$$\le C + 4K/3 + \sqrt{K\gamma C} + 2\sqrt{K}\left(\sqrt{C} + \sqrt{K/3} + \sqrt{\sqrt{K\gamma C}}\right)$$
$$\le C + \left(\frac{4 + 2\sqrt{3}}{3}\right)K + (\gamma^{\frac{1}{2}} + 2)K^{\frac{1}{2}}C^{\frac{1}{2}} + 2\gamma^{\frac{1}{4}}K^{\frac{3}{4}}C^{\frac{1}{4}}$$
$$\le C + 3K + 3K^{\frac{1}{2}}C^{\frac{1}{2}} + 2K^{\frac{3}{4}}C^{\frac{1}{4}}.$$

Since $K^{3/4}C^{1/4} \le \max\{K, K^{1/2}C^{1/2}\}$, we have $K^{3/4}C^{1/4} \le K + K^{1/2}C^{1/2}$. Therefore,

$$Q_i^* \le C + 5K + 5K^{\frac{1}{2}}C^{\frac{1}{2}}.$$

Since $Q_i^{s,*} \le \gamma C_\epsilon + \min(1, d_{m_i})$, we have $C \le C_\epsilon + \min(1/\gamma, d_{m_i}/\gamma)$. Also, since $1 + \gamma \le 2$, we have

$$Q_i^* - C_\epsilon \le \frac{\min(1, d_{m_i})}{\gamma} + 10\frac{1+\gamma}{\gamma}\ln\frac{2}{\delta} + 5\sqrt{2\frac{1+\gamma}{\gamma}\ln\frac{2}{\delta}}\sqrt{C_\epsilon + \frac{1}{\gamma}}$$
$$\le \frac{\min(1, d_{m_i})}{\gamma} + \frac{20}{\gamma}\ln\frac{2}{\delta} + 10\sqrt{\frac{1}{\gamma}\ln\frac{2}{\delta}}\sqrt{C_\epsilon + \frac{1}{\gamma}}.$$

Since $\delta < 1/2$, $\ln(2/\delta) \ge 1$. Since $\gamma \ge 1/C_\epsilon$, the above bound can be written as

$$Q_i^* - C_\epsilon \le \frac{\min(1, d_{m_i})}{\gamma} + (20 + 10\sqrt{2})\ln\frac{2}{\delta}\sqrt{\frac{C_\epsilon}{\gamma}}$$
$$\le \frac{\min(1, d_{m_i})}{\gamma} + 35\ln\frac{2}{\delta}\sqrt{\frac{C_\epsilon}{\gamma}}.$$

□

Note that the Primal Approximation Algorithm always generates feasible dual solutions, therefore, the dual objective is always no bigger than the optimal primal solution by weak duality. We use OPT to denote the primal optimal solution of the relaxed offline formulation. Combing Lemmas 6, 9, and 7. Recall that $q_{i,min} = \min_{\{j:\hat{y}_{i,j}^{s,*}=1\}} q_{i,j}^s$ and $q_{min} = \min_i q_{i,min}$. Since we required $\ln(2/\delta) \ge 1$ and $\gamma \ge 1/C_\epsilon$, we have that

$$P(\hat{\mathbf{y}}^*) = D(\hat{\mathbf{x}}^*, \hat{\mathbf{z}}^*, \hat{\mathbf{t}}^*) + \sum_{i=1}^{n}\left(\sum_{j=1}^{m} q_{i,j}\hat{y}_{i,j}^* - C_\epsilon\right)\hat{x}_i$$
$$\le OPT + \sum_{i=1}^{n}\left(\frac{\min(1, d_{m_i})}{\gamma} + 35\ln\frac{2}{\delta}\sqrt{\frac{C_\epsilon}{\gamma}}\right)\frac{1}{q_{i,min}}$$
$$\le OPT + \frac{1}{q_{min}}\left(\min(m,n) + 35n\ln\frac{2}{\delta}\sqrt{\frac{C_\epsilon}{\gamma}}\right).$$

Since $OPT \ge nC_\epsilon$,

$$P(\hat{\mathbf{y}}^*) \le OPT\left(1 + \frac{\min(m,n)}{q_{min}nC_\epsilon} + \frac{35\ln(2/\delta)}{q_{min}\sqrt{\gamma C_\epsilon}}\right).$$

Finally, we need to bound the prediction error. Recall that we know that

$$C_\epsilon \le \sum_{j=1}^{\gamma m} q_{i,j}^s \hat{y}_{i,j}^{s,*} \le \gamma C_\epsilon + 1,$$

and

$$|\gamma Q_i^* - Q_i^s| \le (1+\gamma)\frac{2}{3}\ln\frac{2}{\delta} + \sqrt{Q_i^* + Q_i^s}\sqrt{2\gamma(1+\gamma)\ln\frac{2}{\delta}}.$$

Hence,

$$Q_i^* \ge C_\epsilon - \left(\frac{1+\gamma}{\gamma}\frac{2}{3}\ln\frac{2}{\delta} + \sqrt{\frac{2(1+\gamma)}{\gamma}\ln\frac{2}{\delta}}\sqrt{Q_i^* + \gamma(C_\epsilon + \frac{1}{\gamma})}\right)$$
$$= C_\epsilon - K/3 - \sqrt{K}\sqrt{Q_i^*} - \sqrt{K}\sqrt{\gamma C}$$

Again, by simple manipulation,

$$Q_i^* + \sqrt{K}\sqrt{Q_i^*} - C_\epsilon + K/3 + \sqrt{K}\sqrt{\gamma C} \ge 0.$$

Using the quadratic formula again,

$$\sqrt{Q_i^*} \le \frac{-\sqrt{K} - \sqrt{K + 4C_\epsilon - 4K/3 - 4\sqrt{K}\sqrt{\gamma C}}}{2}$$

or

$$\sqrt{Q_i^*} \ge \frac{-\sqrt{K} + \sqrt{K + 4C_\epsilon - 4K/3 - 4\sqrt{K}\sqrt{\gamma C}}}{2}.$$

Note that the first inequality never holds since $\sqrt{Q_i^*} \ge 0$. Squaring both sides of the second inequality,

$$Q_i^* \ge \frac{K}{4} + \frac{K}{4} + C_\epsilon - \frac{K}{3} - \sqrt{K}\sqrt{\gamma C}$$
$$- \frac{1}{2}\sqrt{K}\sqrt{K + 4C_\epsilon - 4K/3 - 4\sqrt{K}\sqrt{\gamma C}}$$
$$\ge C_\epsilon + \frac{K}{6} - \sqrt{\gamma CK}$$
$$- \frac{1}{2}\sqrt{K}(\sqrt{K} + 2\sqrt{C_\epsilon} - 2\sqrt{K/3} - 2\sqrt{\sqrt{K}\sqrt{\gamma C}})$$
$$\ge C_\epsilon + 0.24K - (\gamma^{1/2} + 1)(CK)^{1/2} + (\gamma C)^{1/4}K^{3/4}$$
$$\ge C_\epsilon - 2K^{\frac{1}{2}}C^{\frac{1}{2}}$$
$$= C_\epsilon - 2\sqrt{2\frac{1+\gamma}{\gamma}\ln\frac{2}{\delta}}\sqrt{C_\epsilon + \frac{1}{\gamma}}$$
$$\ge C_\epsilon - 4\sqrt{\frac{1}{\gamma}\ln\frac{2}{\delta}}\sqrt{C_\epsilon + \frac{1}{\gamma}}.$$

Again, since $\delta < 1/2$ and $\gamma \geq 1/C_\epsilon$,

$$\begin{aligned} Q_i^* &\geq C_\epsilon - 4\sqrt{2}\sqrt{\frac{1}{\gamma}\ln\frac{2}{\delta}}\sqrt{C_\epsilon} \\ &= C_\epsilon(1 - 4\sqrt{2}\sqrt{\frac{1}{\gamma}\ln\frac{2}{\delta}}\frac{1}{\sqrt{C_\epsilon}}) \\ &\geq C_\epsilon(1 - 6\ln\frac{2}{\delta}\frac{1}{\sqrt{\gamma C_\epsilon}}). \end{aligned} \tag{10}$$

Therefore, if labels are aggregated using weighted majority voting, the prediction accuracy is bounded by ϵ', where

$$\epsilon' = \epsilon^{1-6\ln(2/\delta)/\sqrt{\gamma C_\epsilon}}$$

because $C_{\epsilon'} = C_\epsilon(1 - 6\ln(2/\delta)/\sqrt{\gamma C_\epsilon})$. □

A.5. Proof of Lemma 3

We have assumed we have an estimate $\hat{p}_{i,j}$ such that $|p_{i,j} - \hat{p}_{i,j}| = \alpha$ for some $\alpha \in [0, t]$. Define $\hat{q}_{i,j} = (2\hat{p}_{i,j} - 1)^2$. We have

$$\begin{aligned} |q_{i,j} - \hat{q}_{i,j}| &= \left|(2p_{i,j} - 1)^2 - (2\hat{p}_{i,j} - 1)^2\right| \\ &= 4\left|p_{i,j}^2 - p_{i,j} - \hat{p}_{i,j}^2 + \hat{p}_{i,j}\right| \\ &= 4\left|(p_{i,j}^2 - \hat{p}_{i,j}^2) - (p_{i,j} - \hat{p}_{i,j})\right| . \end{aligned} \tag{11}$$

Consider the two terms in the absolute value in Equation 11. First note that these two terms always have the same sign; both are positive if $p_{i,j} > \hat{p}_{i,j}$, negative if $p_{i,j} < \hat{p}_{i,j}$, and 0 if $p_{i,j} = \hat{p}_{i,j}$. Therefore, we can write

$$\begin{aligned} |q_{i,j} - \hat{q}_{i,j}| &= 4\left|\left|p_{i,j}^2 - \hat{p}_{i,j}^2\right| - |p_{i,j} - \hat{p}_{i,j}|\right| \\ &= 4\left|\left|p_{i,j}^2 - \hat{p}_{i,j}^2\right| - \alpha\right| . \end{aligned}$$

Consider the case in which $p_{i,j} \geq \hat{p}_{i,j}$. Then

$$\begin{aligned} \left|p_{i,j}^2 - \hat{p}_{i,j}^2\right| &= p_{i,j}^2 - \hat{p}_{i,j}^2 \leq p_{i,j}^2 - (p_{i,j} - \alpha)^2 \\ &= 2\alpha p_{i,j} - \alpha^2 \leq 2\alpha . \end{aligned}$$

A symmetric argument can be made for the case in which $p_{i,j} < \hat{p}_{i,j}$. So $\left|p_{i,j}^2 - \hat{p}_{i,j}^2\right| \in [0, 2\alpha]$ and therefore

$$|q_{i,j} - \hat{q}_{i,j}| \leq 4\alpha \leq 4t.$$

□

A.6. Proof of Theorem 3

We first show that the optimal solution of the relaxed offline formulation with the true $q_{i,j}$ values and parameter ϵ is feasible in the approximated LP using the values $\hat{q}_{i,j} = (2\hat{p}_{i,j} - 1)^2$ and parameter ϵ'. Let $\bar{\mathbf{y}}^*$ be the optimal solution of the relaxed offline formulation. To show that $\bar{\mathbf{y}}^*$ is feasible in the approximated LP, we need only show that for every task i, $\bar{\mathbf{y}}^*$ satisfies

$$\sum_{j=1}^m \hat{q}_{i,j}\bar{y}_{i,j}^* \geq C_{\epsilon'} = 2\ln\left(\frac{1}{\epsilon^{1-4t/\bar{q}_{min}^*}}\right), \tag{12}$$

because $\hat{\mathbf{q}}$ and $C_{\epsilon'}$ only appear in this constraint.

Fix a task i. Since we assumed that for every j, $|p_{i,j} - \hat{p}_{i,j}| \leq t$, we have from Lemma 3 that $\hat{q}_{i,j} \geq q_{i,j} - 4t$ for all j, and

$$\sum_{j=1}^m \hat{q}_{i,j}\bar{y}_{i,j}^* \geq \sum_{j=1}^m q_{i,j}\bar{y}_{i,j}^* - 4t\sum_{j=1}^m \bar{y}_{i,j}^*.$$

Since $\bar{\mathbf{y}}^*$ is optimal in the relaxed offline formulation we can easily show that $\Sigma_{j=1}^m q_{i,j}\bar{y}_{i,j}^* = C_\epsilon$ for all i. Assume for contradiction that this was not the case. We cannot have $\Sigma_{j=1}^m q_{i,j}\bar{y}_{i,j}^* < C_\epsilon$ for any i since this would violate the constraints of the relaxed offline formulation. Assume that $\Sigma_{j=1}^m q_{i,j}\bar{y}_{i,j}^* > C_\epsilon$ for some i. Then we could always decrease some $y_{i,j}^*$ of task i to make the equality hold, which would decrease the objective value without violating any other constraint, meaning that the solution could not be optimal.

By definition of $\bar{q}_i^*$, i.e., $\bar{q}_i^* = \Sigma_{j=1}^m q_{i,j}\bar{y}_{i,j}^*/\Sigma_{j=1}^m \bar{y}_{i,j}^*$, we also can get that $\Sigma_{j=1}^m \bar{y}_{i,j}^* = C_\epsilon/\bar{q}_i^*$.

Therefore,

$$\sum_{j=1}^m \hat{q}_{i,j}\bar{y}_{i,j}^* \geq C_\epsilon - 4tC_\epsilon/\bar{q}_i^* = C_\epsilon(1 - 4t/\bar{q}_i^*).$$

Since $C_\epsilon = 2\ln(1/\epsilon)$,

$$\begin{aligned} C_\epsilon(1 - 4t/\bar{q}_i^*) &= 2(1 - 4t/\bar{q}_i^*)\ln(1/\epsilon) \\ &= 2\ln\left(\frac{1}{\epsilon^{1-4t/\bar{q}_i^*}}\right). \end{aligned}$$

Hence,

$$\sum_{j=1}^m \hat{q}_{i,j}\bar{y}_{i,j}^* \geq 2\ln\left(\frac{1}{\epsilon^{1-4t/\bar{q}_i^*}}\right) \geq 2\ln\left(\frac{1}{\epsilon^{1-4t/\bar{q}_{min}^*}}\right).$$

which shows that $\bar{\mathbf{y}}^*$ satisfies Equation 12 for all i.

Since the optimal solution of the relaxed offline formulation is feasible in the approximated LP, the optimal objective of the approximated LP is no bigger than the optimal objective of the relaxed offline formulation. Furthermore, the optimal objective of the relaxed offline formulation is always no bigger than the optimal objective of the offline integer program. Therefore, the optimal objective of the approximated LP would be no bigger than the offline optimal of integer program. □

A.7. Proof of Theorem 4

We begin by restating the main definitions from the theorem statement for easy reference. Recall that $\bar{q}^*_{min}$ is a value such that $\bar{q}^*_{min} \leq \bar{q}^*_i$ for all i, and our estimates $\hat{p}_{i,j}$ are guaranteed to satisfy $|p_{i,j} - \hat{p}_{i,j}| \leq t$ for all (i,j) pairs for some $t < \bar{q}^*_{min}/4$. Recall that $\mathbf{y}$ is any feasible integer assignment of the approximated LP with parameter $\epsilon' = \epsilon^{1-4t/\bar{q}^*_{min}}$ and skill levels $\hat{q}_{i,j} = (2\hat{p}_{i,j} - 1)^2$, $J_i = \{j : y_{i,j} = 1\}$ denotes the set of workers that are assigned to task i according to the feasible integer assignment, and $\hat{q}_i = \Sigma_{j \in J_i} \hat{q}_{i,j} / |J_i|$. Tasks are assigned according to $\mathbf{y}$ and the results are aggregated using weighted majority voting with weights $w_{i,j} = 2\hat{p}_{i,j} - 1$, so

$$\hat{\ell}_i = \text{sign}\left(\sum_{j \in J_i} (2\hat{p}_{i,j} - 1)\ell_{i,j}\right).$$

Below we show that $\hat{\ell}_i = \ell_i$ with probability at least $1 - \epsilon^{(1-4t/\bar{q}^*_{min})(1-4t/\hat{q}_{min})}$.

Without loss of generality, assume that $\ell_i = 1$. Note that although we weight each label by $2\hat{p}_{i,j} - 1$, the probability of label $\ell_{i,j}$ being correct (i.e., the probability that $\ell_{i,j} = \ell_i = 1$) is still $p_{i,j}$. Similar to the proof of Lemma 1, let

$$X_{i,j} = \begin{cases} 2\hat{p}_{i,j} - 1 & \text{with probability } p_{i,j} \\ -2\hat{p}_{i,j} + 1 & \text{with probability } 1 - p_{i,j}, \end{cases}$$

and

$$X_i = \sum_{j \in J_i} X_{i,j}.$$

Let $t_{i,j} = p_{i,j} - \hat{p}_{i,j}$. Then

$$\begin{aligned} E[X_i] &= \sum_{j \in J_i} E[X_{i,j}] \\ &= \sum_{j \in J_i} (p_{i,j}(2\hat{p}_{i,j} - 1) + (1 - p_{i,j})(-2\hat{p}_{i,j} + 1)) \\ &= \sum_{j \in J_i} (4p_{i,j}\hat{p}_{i,j} - 2p_{i,j} - 2\hat{p}_{i,j} + 1) \\ &= \sum_{j \in J_i} (4(\hat{p}_{i,j} + t_{i,j})\hat{p}_{i,j} - 2(\hat{p}_{i,j} + t_{i,j}) - 2\hat{p}_{i,j} + 1) \\ &= \sum_{j \in J_i} (2\hat{p}_{i,j} - 1)^2 + \sum_{j \in J_i} 2t_{i,j}(2\hat{p}_{i,j} - 1) \\ &\geq \sum_{j \in J_i} (2\hat{p}_{i,j} - 1)^2 - \sum_{j \in J_i} 2\,|t_{i,j}|\,|2\hat{p}_{i,j} - 1|. \end{aligned}$$

The first term is simply $|J_i|\,\hat{q}_i$. Since, by assumption, $|t_{i,j}| \leq t$ for all (i,j), and since $|2\hat{p}_{i,j} - 1| \leq 1$ for all (i,j), we have

$$E[X_i] \geq |J_i|\,\hat{q}_i - 2t\,|J_i| = |J_i|\,(\hat{q}_i - 2t).$$

Similar to the proof of Lemma 1, the probability of predicting the wrong label for task i is bounded by $\Pr(X_i \leq 0) = \Pr(E[X_i] - X_i \geq E[X_i])$. Using Hoeffding's inequality and the bound above, we have

$$\begin{aligned} Pr(X_i \leq 0) &\leq \exp\left(-\frac{2(E[X_i])^2}{\sum_{j \in J_i}(4\hat{p}_{i,j} - 2)^2}\right) \\ &\leq \exp\left(-\frac{2|J_i|^2(\hat{q}_i - 2t)^2}{4|J_i|\hat{q}_i}\right) \\ &= \exp\left(-\frac{|J_i|(\hat{q}_i - 2t)^2}{2\hat{q}_i}\right) \\ &= \exp\left(-\frac{|J_i|\hat{q}_i{}^2(1 - 2t/\hat{q}_i)^2}{2\hat{q}_i}\right) \\ &= \exp\left(-\frac{1}{2}|J_i|\hat{q}_i(1 - 2t/\hat{q}_i)^2\right) \\ &\leq \left(\epsilon^{1-4t/\bar{q}^*_{min}}\right)^{(1-2t/\hat{q}_i)^2} \\ &= \epsilon^{(1-4t/\bar{q}^*_{min})(1-2t/\hat{q}_i)^2} \\ &\leq \epsilon^{(1-4t/\bar{q}^*_{min})(1-4t/\hat{q}_i)}. \end{aligned}$$

The sixth line follows from the constraints of the approximated LP which require that

$$|J_i|\hat{q}_i = \sum_{j \in J_i} \hat{q}_{i,j} \geq 2\ln(1/\epsilon'),$$

and so $\exp(-|J_i|\hat{q}_i/2) \leq \epsilon'$. □

A.8. Proof of Theorem 5

The proof is a direct application of Lemma 2 and Theorems 2, 3, and 4.

We assign each worker s gold standard tasks for each task type. If we use the empirical skill level $\hat{p}_{i,j}$ as an estimate of $p_{i,j}$ as in Lemma 2, we know that for any δ', with probability at least $1 - \delta'$, $|p_{i,j} - \hat{p}_{i,j}| \leq \sqrt{\ln(2/\delta')/(2s)}$.

Note that we have T types of tasks and $(1+\gamma)m$ workers. Hence, we need to assign $T(1+\gamma)ms$ gold standard tasks in total to workers. Replacing δ' with $\delta/2$ and applying the union bound for all types of tasks and all workers, we know that with probability at least $1-\delta/2$,

$$|p_{i,j} - \hat{p}_{i,j}| \leq \sqrt{\frac{\ln(4T(1+\gamma)m/\delta)}{2s}} \text{ for all } (i,j). \tag{13}$$

Let $t = \sqrt{\ln(4T(1+\gamma)m/\delta)/(2s)}$. If we set $\hat{q}_{i,j} = (2\hat{p}_{i,j}-1)^2$, from Lemma 3, we have an estimate $\hat{\mathbf{q}}$ such that with probability at least $1-\delta/2$, $|q_{i,j} - \hat{q}_{i,j}| \leq 4t$ for all i and j. Given $\hat{\mathbf{q}}$ and access to $\bar{q}^*_{min}$ (as assumed in statement of the theorem), we can create the approximated LP as described in Section 5.2 with $\hat{\mathbf{q}}$ and $C_{\epsilon'}$ where $\epsilon' = \epsilon^{1-4t/\bar{q}^*_{min}}$.

Bounding Competitive Ratio

Below we bound the number of non-gold standard tasks assignment of the Main Algorithm. In Section 5.1, we assumed that the $q_{i,j}$ values of worker j are revealed when worker j arrives. However, now we have only estimates of these values. We can still apply Theorem 2, but now with the estimated values $\hat{q}_{i,j}$ in place of the $q_{i,j}$, and ϵ' in place of ϵ. The relaxed online formulation with these replacements is exactly the approximated LP of Section 5.2. Applying Theorem 2 with δ set to $\delta/2$, with probability at least $1-\delta/2$, the Primal Approximation Algorithm with inputs $\hat{\mathbf{x}}$ and $\hat{\mathbf{q}}$ yields an assignment $\mathbf{y}$ such that the number of tasks assigned using $\mathbf{y}$ is no more than

$$\left(1 + \frac{\min(m,n)}{\hat{q}_{min} n C_{\epsilon'}} + \frac{35\ln(4/\delta)}{\hat{q}_{min}\sqrt{\gamma C_{\epsilon'}}}\right)$$

times the optimal value of the approximated LP, where $\hat{q}_{min} = \min_{(i,j):y^s_{i,j}=1} \hat{q}^s_{i,j}$, and $\mathbf{y}^s$ is the primal optimal solution of the sampled LP with the replacements described above.

From Theorem 3, we know that the optimal of the approximated LP yields no more assignment than the optimal of the relaxed offline formulation. Combining the above results, with probability of at least $1-\delta/2$, the Main Algorithm yields no more assignments than

$$\left(1 + \frac{\min(m,n)}{\hat{q}_{min} n C_{\epsilon'}} + \frac{35\ln(4/\delta)}{\hat{q}_{min}\sqrt{\gamma C_{\epsilon'}}}\right)$$

times the optimal of the relaxed offline formulation, which is less than the optimal of the IP.

Since the analysis above holds only if $|p_{i,j} - \hat{p}_{i,j}| \leq t$ for all (i,j) and we know this happens with probability at least $1-\delta/2$, we can apply union bound to show that the competitive ratio above can be achieved with probability of at least $1-\delta$.

Bounding Prediction Error

We now bound the prediction error. By a similar argument to the one used above, we can use the same argument made to achieve the error bound in Theorem 2 (Equation 10) to guarantee that if the high probability events above hold, then

$$\sum_{j=1}^{m} \hat{q}_{i,j} y_{i,j} \geq C_{\epsilon'} \left(1 - 6\frac{\ln(4/\delta)}{\sqrt{\gamma C'_\epsilon}}\right)$$

where $\epsilon'' = \epsilon'^{(1-6\ln(4/\delta)/\sqrt{\gamma C_{\epsilon'}})}$.

Assume that the high probability event from above holds and so $|p_{i,j} - \hat{p}_{i,j}| \leq t$ for all (i,j). Let $\hat{q}_i = \sum_{j:y_{i,j}=1} q_{i,j}/|\{j : y_{i,j} = 1\}|$. Applying Theorem 4 with the value ϵ'' in place of ϵ' (since $\mathbf{y}$ is a feasible assignment to the relaxed LP with ϵ'' used instead of ϵ'), we have that the prediction error of each task i is bounded by

$$\begin{aligned}\epsilon''^{1-4t/\hat{q}_i} &\leq \epsilon'^{(1-6\ln(4/\delta)/\sqrt{\gamma C'_\epsilon})(1-4t/\hat{q}_i)} \\ &\leq \epsilon^{(1-4t/\bar{q}^*_{min})(1-6\ln(4/\delta)/\sqrt{\gamma C'_\epsilon})(1-4t/\hat{q}_i)}.\end{aligned}$$

Replacing s in Equation 13, we have that $|p_{i,j} - \hat{p}_{i,j}| \leq t$ for all i and j where $t \leq \sqrt{\ln(4T(1+\gamma)m/\delta)/(2s)}$. Finally, replacing t in the error bound above completes the proof. □