NYU Stern Research Working Paper

CBA-13-06

Quality-Based Pricing for Crowdsourced Workers

Jing Wang
jwang5@stern.nyu.edu

Panagiotis G. Ipeirotis
panos@stern.nyu.edu

Foster Provost
fprovost@stern.nyu.edu

June 12, 2013

Abstract

The emergence of online *paid crowdsourcing* platforms, such as Amazon Mechanical Turk (AMT), presents us huge opportunities to distribute tasks to human workers around the world, on-demand and at scale. In such settings, online workers can come and complete tasks posted by a company, and work for as long or as little as they wish. Given the absolute freedom of choice, crowdsourcing eliminates the overhead of the hiring (and dismissal) process. However, this flexibility introduces a different set of inefficiencies: verifying the quality of every submitted piece of work is an expensive operation, which often requires the same level of effort as performing the task itself. There are many research challenges that emerge in this paid-crowdsourcing setting. How can we ensure that the submitted work is accurate? How can we estimate the quality of the workers, and the quality of the submitted results? How should we pay online workers that have imperfect quality? We present a comprehensive scheme for managing quality of crowdsourcing processes: First, we present an algorithm for estimating the quality of the participating workers and, by extension, of the generated data. We show how we can separate systematic worker biases from unrecoverable errors and how to generate an unbiased "worker quality" measurement that can be used to objectively rank workers according to their performance. Next, we describe a pricing scheme that identifies the fair payment level for a worker, adjusting the payment level according to the contributed information by each worker. Our pricing policy, which pays workers based on their expected quality, reservation wage, and expected lifetime, estimates not only the payment level but also accommodates measurement uncertainties and allows the workers to receive a fair wage, even in the presence of temporary incorrect estimations of quality. Our experimental results demonstrate that the proposed pricing strategy performs better than the commonly adopted uniform-pricing strategy. We conclude the paper by describing strategies that build on our quality control and pricing framework, to build crowdsourced tasks of increasingly higher complexity, while still maintaining a tight quality control of the process, even if we allow participants of unknown quality to join the process.

1 Introduction

Crowdsourcing has emerged over the last few years as an important new labor pool for a variety of tasks (Malone et al., 2010), ranging from micro-tasks on Amazon Mechanical Turk to big innovation contests conducted by Netflix and Innocentive. Amazon Mechanical Turk (AMT) today dominates the market for crowdsourcing micro-tasks that are trivial to humans, but challenging to computer programs (Ipeirotis, 2010). The requesters can post tasks, such as image tagging, language translation, event annotation, and workers complete them and get compensated in the form of micro-payments (Snow et al., 2008). The immediate availability of labor supply makes it possible to start completing these tasks with very low latency and with high throughput.

Despite the promise, significant challenges remain. Workers in the crowdsourcing markets usually have different expertise, background and incentives; therefore, they are likely to exhibit heterogeneous quality in their submitted work. Unfortunately, verifying the quality of every submitted answer is an expensive operation and negates many of the advantages of crowdsourcing: the cost and time for verifying the correctness of the submitted answers is typically comparable to the cost and time for performing the task itself. The difficulty of verification leads many workers to be less worried about submitting perfect work, as there is a high probability that incorrect submissions may not be checked. The lax supervision, combined with the common pricing scheme of uniform pricing (i.e., paying all the workers the same price for completing the same task), generates unfortunate incentives: crowdsourced tasks are more appealing to workers who exhibit lower quality, both those who don't have the required skills and those who invest very little effort. This is known as "the bad driving out the good" in the market (Akerlof, 1970). The abundance of low-quality workers undoubtedly harms the scalability and robustness of online markets.

One commonly known approach for dealing with this problem is to use "gold" data: To perform some form of quality control, employers often insert a small percentage of tasks for which they already know the correct answers, and measure the performance against these tasks. Such a setting reduces the task of performance monitoring into a problem that could in principle be handled by test theory (Crocker and Algina, 2006; DeMars, 2010). With this setting, we can measure the quality of the workers, and eliminate from the workforce the underperforming workers.

Another method to ensure quality is to rely on majority voting: simply ask multiple workers to complete the same task and use majority voting to identify the correct answers. In that case, we do not estimate the quality of the workers but instead try to generate a work output that is of high quality. In reality, most employers check agreement of workers with majority vote and dismiss workers that are systematically in disagreement with the majority. A negative aspect of this approach is that workers of low quality can still participate during the data collection phase, driving down the signal-to-noise ratio, and lowering the quality of the results for a given level of monetary investment.

In this paper, we propose a hybrid scheme that combines both approaches for measuring quality. We extend the expectation maximization framework introduced by Dawid and Skene (1979), and we create a flexible framework that can use any combination of redundancy and gold testing to jointly estimate the correct answer for each task and a "quality score" for each worker and data point. The quality score is an *unbiased estimate* of the true uncertainty in the answers of the worker, after removing any systematic bias. Similarly, for a data point, the quality score is an estimate of the remaining uncertainty about the correctness of the computed answer.

Given a reliable method for estimating the quality of the workers, we then turn our attention to determining a fair pricing scheme for the workers. We start by determining the price for a worker who meets the standards of quality set by the employer; then show how to compute a fair price for a worker who does not meet the quality standards set by the employer. As quality measurements are inherently uncertain, we also establish a payment scheme in which we pay workers based on the lower estimate of their quality, essentially withholding some payment for those who really are the better workers. However, as our quality estimates become more certain over time, we refund the "withheld" payment, ensuring that, in the limit, we give to workers a payment that corresponds to their true quality, even in the presence of measurement uncertainties.

In our work, we focus on quality control for tasks that have answers consisting of a small set of discrete choices (e.g., "Does this photo violate the terms of service? Yes or No."). While this may seem limiting, we show in Section 3 that many complex tasks can performed by breaking them down into a set of simple operations. Our proposed scheme naturally fits into such workflows and provides the a fundamental quality control block, which in turn allows for quality control of other operations. Such synergies lead to workflows that can complete complex tasks with guarantees of high-quality output, even when the underlying workforce has uncertain, varying, or even moderate-to-low quality.

The rest of the paper is structured as follows: First, in Section 2, we review related work. Then, in Section 3, we describe how to structure complex tasks, using a set of fundamental components, in order to allow for quality-controlled execution of complex tasks using crowdsourced labor. Section 4 describes our setting in a more formal manner, outlines the assumptions of the model, and gives a simple pricing model for pricing workers with quality above the requirements for the task. In Section 5, we proceed to describe our quality-estimation framework, which estimates the quality of the worker based on the estimated cost of the *unrecoverable* mistakes that the worker is expected to make while performing a task. In Section 6, we extend our approach to deal with a "streaming" environment, where workers and labels arrive over time, while we are running the task, and we need to deliver completed work, as soon as it is ready. Section 7 uses the quality estimation results to propose a pricing scheme that rewards workers according to their quality and the competition in the market, relaxing the assumption that all workers satisfy the quality requirements. Our experimental results in Section 8 demonstrate a significant improvement over existing baselines, both in terms of data quality but also in terms of workforce engagement. We conclude by describing the managerial implications and directions for future research.

2 Related Work

2.1 The Market for "Lemons"

When there exist both quality heterogeneity and asymmetric information, the market could easily become a market for "lemons". In his classic paper, Akerlof (1970) uses the market for used cars as an example to show that asymmetrically held information can lead to market inefficiency. Since buyers cannot tell the difference between a good car and a bad car, all cars are sold at the same price based on the quality of the average used car in the market. Good car owners, having knowledge of the high quality of their cars, will not place their cars on the market. The withdrawal of good cars will then reduce the average quality of cars sold on the market, as well as the buyers' willingness to pay. This process might continue until we end up with only "lemons" in the market.

In crowdsourcing, due to the relative anonymity of the "crowd" workers, the employer who uses crowdsourcing cannot readily assess the credentials and quality differences among workers. If workers are paid at a single price (a common practice today), and employers are adjusting their prices to accommodate for the cost of dealing with low-quality workers, we get a setting where good workers are leaving the market, and only low-quality workers remain. This might cause market failure, that is, "it is quite possible to have the bad driving out the not-so-bad driving out the good in such a sequence of events that no market exists at all" (Akerlof, 1970). Arguably, the lack of a well-designed reputation and signaling mechanism led to such phenomena in the Amazon Mechanical Turk market, forcing Amazon create a two-class reputation for the workers: the "Masters" workers (approximately 2% of the workers) that have "proven" themselves in the market, and the rest. As easily imagined, there is significant room for improvement.

When speaking of the design of contracts under asymmetric information, it is natural to consider the literature on costly state verification (CSV) (Townsend, 1979; Gale and Hellwig, 1985) in which there exists some cost for revealing information that would otherwise be private. The main result of the CSV approach is that it is generally optimal to commit to state-contingent (ex post) verification strategy: Verification occurs if and only if the realization of endowment falls in the verification region. As we will describe in more detail later, the assumptions of the CSV are not directly applicable in crowdsourcing markets. First, CSV assumes that verification is perfect when it occurs, but the revealing of worker quality usually involves continuous testing, with noticeable errors especially in the early stage. Second, CSV associates a positive cost with verification; however, as we will show later, there exist almost costless verification in crowdsourcing settings. In an exploration-exploitation tradeoff situation, we use redundancy and allow imperfect labels to be used for worker quality verification (exploration), while at the same time we use the signal of the worker quality to label our data (exploitation).

Another mechanism to deal with this problem is reputation systems (Resnick et al., 2000). For example, the past performance of a worker informs us about her true quality. This in turn, allows a better pricing model for workers with different levels of qualities. Employers who have past interactions with workers have the ability to learn the quality of the workers, and in turn, build their own, private reputation systems. Our work works in tandem with the existence of a reputation system. First, our work shows how to measure objectively the quality of the work submitted by different workers, while keeping the costs of verification low. Second, the prior knowledge of worker reputation can effortlessly be included in our system as a prior belief about the worker quality, and then our proposed model can update the belief as the worker engages in the current task.

2.2 Quality Estimation

To measure the quality of the workers, we can insert some tasks with "gold" labels (i.e., tasks for which we know the correct answer) into the stream of assigned tasks, and then compute the error rate of each worker to detect the spammers. The testing of worker quality using "gold" labels is related to two lines of research: test theory in psychometrics and education, and acceptance sampling in operation management.

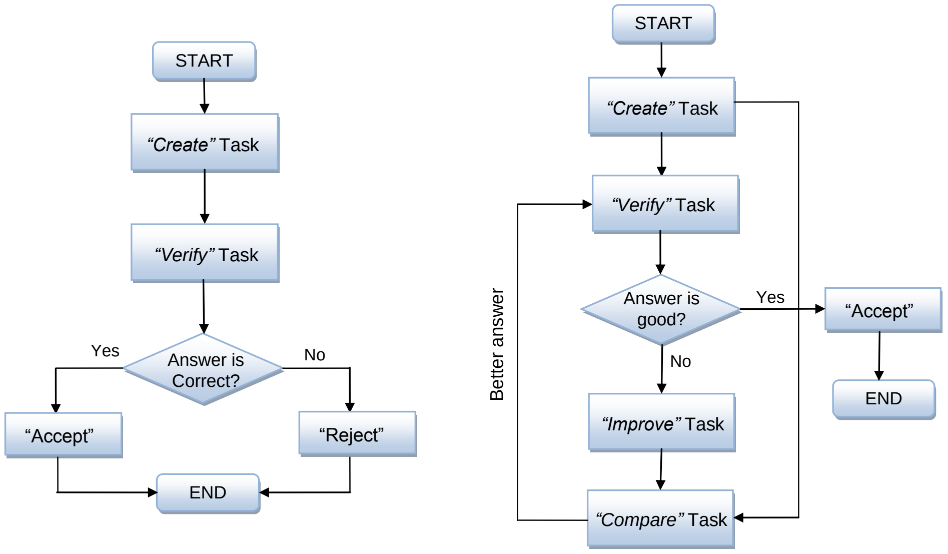
Classical test theory (CTT) (Crocker and Algina, 2006) is a widely known measurement approach to measuring individual differences. CTT assumes that each observed test score consist of two components: a *true score* and an *error score*. While CTT has been successfully applied in test development, some important shortcomings have been recognized: First, the two classical statistics, item difficulty and item discrimination, are both sample dependent. Second, comparison of test-takers are limited to situations in which they take the same form of the test. Item response theory (IRT) (DeMars, 2010) overcomes limitations of CTT by assuming a single latent variable representing the ability of each examinee. The probability of a correct item response does not only depend on the individual ability, but also on item characteristics (e.g., item difficulty, item discrimination). Nowadays, with the increasing availability of Internet and personal computers, computerized adaptive testing (CAT) becomes feasible. CAT allows that the set of items that each examinee receives can be unique, adapting to the examinee's ability level.

The set of test theory models discussed above is good in terms of accurate ability estimation, however, these models do not take into consideration the additional costs that can be incurred. In crowdsourcing markets, each time we test a worker, we forfeit the opportunity to get some work done. This is analogous to the concept of inspection cost in manufacturing process. There has been a tremendous amount of work on the topic of optimal acceptance sampling plans in quality control (Dodge and for Quality Control, 1973; Wetherill and Chiu, 1975; Berger, 1982; Schilling, 1982). The purpose of acceptance sampling is to determine whether to accept or reject a production lot of items, by selecting a sample for inspection. Optimal acceptance sampling maximizes the profits of producers by striking the appropriate balance between quality assurance and total cost. A key difference of our work is: in acceptance sampling, a production lot of items will get rejected if the number of defective items in a sample exceeds a threshold; whereas in crowdsourcing markets that deal with information goods, low-quality work can be combined to provide high-quality outcomes.

Dawid and Skene (1979) presented an expectation maximization (EM) algorithm to estimate the error rates of observers when all observers see all available patients. Other versions of the algorithm were recently proposed by Raykar et al. (2010) and by Carpenter (2008). The algorithm iterates until convergence, following two steps: (1) estimates the true response for each patient, using records given by all the observers, accounting for the error-rates of each observer; and (2) estimates the error-rates of observers by comparing the submitted records to estimated true response. The final output of the EM algorithm is the estimated true response for each patient and estimated error-rate represented by "confusion matrix" for each observer. Whitehill et al. (2009) presented a probabilistic model to simultaneously infer the label of each image, the expertise of each labeler, and the difficulty of each image. Their assumption is that the log odds for the obtained labels being correct are a bilinear function of the difficulty of the label and the expertise of the labeler. Welinder et al. (2010) proposed a generative Bayesian model in which each annotator is a multidimensional entity with variables representing competence, expertise and bias. They also described an inference algorithm to estimate the properties of the data being labeled and the annotators labeling them. Bachrach et al. (2012) proposed a probabilistic graphical model to jointly infer the correct answer and difficulty level for each question, and the ability of each participant. Moreover, they test the ability level of participant in an adaptive way, similarly to the approach used in CAT. The common objective across all the approaches above is to estimate the "confusion matrix" for each worker. In our work, we leverage the confusion matrix information to understand the true quality of a worker, eliminate biases, and define a payment scheme that incentivizes workers to work at high-quality levels, discouraging worker churn at the same time.

2.3 Payment Schemes

The choice of payment schemes has always been a central topic in labor economics. A variety of payment schemes have been proposed and used in practice, such as flat-rate, piece-rate, tournament (Lazear and Rosen, 1979), and relative piece-rate (Mookherjee, 1984; Meyer and Vickers, 1997). The effectiveness of flat rate (fixed payment) partially relies on the long-term relationship between organizations and their employees, which is not suitable in the online setting since the contract is formed on a temporary basis. Piece-rate scheme would have unfortunate consequences even under



(a) Correct vs. incorrect answers

(b) Varying degrees of correctness

Figure 1: Workflows for two different types of tasks

effective verification: Workers with heterogeneous quality levels are paid the same as long as they meet the prescribed quality level, which is in general far from the one promised in SLA. Tournament and relative piece-rate pay workers based on the relative ranking and relative performance of workers, respectively. Some empirical studies have attempted to assess the relative effectiveness of different payment schemes (Lazear, 1986, 2000; Agranov and Tergiman, 2012). But most of these studies measure the performance of workers in terms of quantity of output rather than quality. In our work, we propose a novel payment scheme in which the compensation of each worker is proportional to the amount of value that she contributes. When there is uncertainty in worker quality estimation, we pay them in a conservative way at first, and reimburse them as the estimation gets more and more accurate.

3 Importance of Quality Control for Multiple Choice Items

Our scheme can be directly applied to multiple choice questions, which already captures a large number of tasks that are crowdsourced today (e.g., image moderation, spam detection, restaurant rating, etc.). We would like to stress, though, that quality control mechanisms for multiple choice questions are in the heart of many other, more complex, tasks that are also executed in crowdsourcing platforms. Below we give some representative examples:

- **Correct vs. incorrect answers**: Consider the task that asks workers to collect information about a given topic; for example, "collect URLs that discuss massive online education courses and their impact on MBA programs." For this type of task, it is usually hard or impossible to enumerate all the correct answers, therefore it is not possible to control the quality of the task using quality control for multiple choice answers directly. However, once an answer is provided, we can easily check its correctness, *by instantiating another task*, asking a binary choice question: "Is this submitted URL about massive online education courses and their impact on MBA programs?" Thereby, we break the task into two tasks: The *"Create"* task, in which one or more workers submit free-form answers, and a *"Verify'* task, in which another set of workers vets the submitted answers, and classifies them as either "correct" or "incorrect". Figure 1(a) illustrates the structure: the "Verify" task controls the quality of the "Create" task; the quality of the "Verify" task is then controlled using a quality control mechanism for multiple choice questions, similar to the one that we present in this paper.

- **Varying degrees of correctness**: There are some tasks whose free-form answers are not right or wrong but have different degrees of correctness or goodness (e.g., "generate a transcript from this manuscript," "describe and explain the image below in at least three sentences"). In such a setting, treating the submitted answers as "correct" or "incorrect" may be inefficient: a rejected answer would be completely discarded, while it is often possible to leverage the low-quality answers to get better results, by simply iterating. Past work (Little et al., 2010) has shown the superiority of the iterative paradigm by demonstrating how workers were able to create image descriptions of excellent quality, even though no single worker put any significant effort in the task. Figure 1(b) illustrates the iterative process. There are four subtasks: The *"Create"* task, in which free-form answers are submitted, the *"Improve"* task, in which workers are asked to improve an existing answer, the *"Compare"* task, in which workers are required to compare two answers and select the better one, and the *"Verify"* task, in which workers decide whether the quality of the answers[1] are good. In this case, the "Compare" and "Verify" are multiple choice tasks, and we can use the mechanisms we present to control the quality of the submitted answers (and of the participating workers). In turn, the "Create" and "Improve"

[1] "Verify" task either accepts input directly from the "Create" task or gets the better answer returned by "Compare" task.

tasks are controlled by the "Verify" and "Compare" tasks, as we can measure the probability that a worker submits an answer of high quality, or the probability that a worker will be able to improve an existing answer.

- **Complex tasks using workflows**: Initial applications of paid crowdsourcing focused mainly on simple and routine tasks. However, many tasks in our daily life are much more complicated (e.g., "proofread the following paragraph from the draft of a student's essay," "write a travel guide about New York City") and recently, there is an increasing trend to accomplish such tasks by dividing complex tasks into a set of microtasks, using workflows. For example, Bernstein et al. (2010) introduced the *"Find-Fix-Verify pattern"* to split text editing tasks into three simple operations: find something that needs fixing, fix the problem if there is one, verify the correctness of the fix. Again, this task ends up having quality control through a set of multiple choice tasks (verification of the fix, verification that something needs fixing). In another cases, Kittur et al. (2011) described a framework for parallelizing the execution of such workflows and Kulkarni et al. (2011) move a step further by allowing workers themselves to design the workflow. As in the case of other tasks that are broken down to workflows of micro-tasks, the quality of these complex tasks can be guaranteed by applying our quality control scheme to each single micro-task, following the paradigms described above.

4 Problem setting

So far, we have described the central role of multiple choice tasks in crowdsourcing tasks. Now, we turn our attention to our modeling assumptions and formalization of the problem. Table 1 summarizes the key notations used in this and subsequent sections.

4.1 Modeling Assumptions

Task: In our labeling task, each object o is associated with a *latent* true class label $T^{(o)}$, picked from one of the L different labels. The true class label $T^{(o)}$ is unknown and the task for workers is to identify the true label for the object o.

Client: The client is the owner of the unlabeled objects, and wants to have the objects labeled with

Notation	Definition
O	The set of objects that need to be labeled
L	The set of possible labels for the objects in O
$T^{(o)}$	True class of object (o)
$\boldsymbol{\pi}$	Vector with prior probabilities for object classes
π_i	Prior probability for class i
$\mathbf{p}^{(o)}$	Vector with probability estimates for the true label of object (o)
$p_i^{(o)}$	Probability that the true label of object (o) is i
$K^{(o)}$	Set of workers that assign labels to object (o)
$O^{(k)}$	Set of objects labeled by worker (k)
$\pi_j^{(k)}$	Probability that worker (k) assigns label j
$l_{(o)}^{(k)}$	Label that worker (k) assigns to object (o)
$I(l_{(o)}^{(k)} = i)$	Indicator function for the event $l_{(o)}^{(k)} = i$
$\mathbf{e}^{(k)}$	Confusion matrix for worker (k)
$e_{ij}^{(k)}$	Probability that worker (k) will classify an object with true category i into category j
$\mathbf{c}$	Matrix with the misclassification costs
c_{ij}	Cost incurred when an object with true label i is classified into category j
τ_c	Cost threshold specified in service level agreement (SLA)
S	Fixed price charged to the client for every object with misclassification cost below τ_c
$w^{(k)}$	Reservation wage of worker (k)
$t^{(k)}$	Lifetime of worker (k)
r^*	Optimal price paid to a qualified worker
$v(\mathbf{e})$	Value of a worker with confusion matrix $\mathbf{e}$

Table 1: Key Notations Used in This Paper

their correct categories. To quantify the quality of labeling, the client provides a set of misclassification costs $\mathbf{c}$; the cost c_{ij} is incurred when an object with true label i is classified into category j. The client requires a service-level agreement (SLA), with the guarantee that the average misclassification cost of the labeling will be lower than a threshold τ_c.[2] The client offers to the platform an exogenously defined, fixed price S for each labeled object[3] with misclassification cost below τ_c.

Platform: The platform is the place for executing the task. *The goal of the platform is to optimize its own rate of profit.* The platform receives, from the outside client, the stream of jobs that need to be completed, together with the quality/cost requirement. The received tasks are posted on the crowdsourcing market for workers to work on. Then, the platform announces a price scheme and pays each worker according to the worker's quality, on a piecemeal (i.e., per task) basis. The platform acts as an intermediary between clients and workers, analogous to the notation of firm by Spulber (1996).

[2] The cost can be determined post-hoc, for example, using acceptance sampling (Schilling, 1982), and determine whether the promised labeling quality was met or not.

[3] While we assume that the price is exogenously defined, the price can also be defined by the platform in response to competitive pressures. The only assumption that we need is the existence of a piece-wise price S for which the good is sold.

One important function of firms is to act as *intermediaries* between buyers and sellers. In particular, firms gather demand and supply information to determine the profit-maximizing prices, serve as guarantors of the product quality, and supervise the suppliers for their customers. The platform we define here plays a similar role: e.g., sets optimal prices to maximize its own profit, ensures a certain level of data quality, and monitor the crowdsourced workers for the clients.

Workers: Workers in crowdsourcing markets come to work on the available tasks. We model each worker (k) with: (1) a *latent* "confusion matrix" $\mathbf{e}^{(k)}$, with $e_{ij}^{(k)}$ being the probability that worker (k) will classify an object with true category i into category j (this confusion matrix captures the quality of the worker); (2) a reservation wage $w^{(k)}$ which is the lowest wage for which the worker will accept to work on the task; and (3) a lifetime $t^{(k)}$ which represents the number of tasks that the worker is willing to work on.[4] The *distribution* $f_{\mathbf{E},W,T}(\mathbf{e}, w, t)$ of qualities, reservation wage, and lifetime is common knowledge. However, the individual values of $\mathbf{e}^{(k)}$, $w^{(k)}$, and $t^{(k)}$ for each worker are all private knowledge of the workers and not known apriori to the platform.

4.2 A Simple Pricing Model for Qualified Workers

Given the above setting, we now present a simple pricing model, which relies on assumptions that we will relax later in the paper. First, we assume that we have perfect knowledge of the internal quality $q^{(k)} = g(\mathbf{e}^{(k)})$ of each worker.[5] Therefore, we can divide the workers into two groups, qualified and not qualified: A worker is a *qualified worker* if the quality of the worker satisfies the service-level agreement; otherwise, the worker is considered as an *unqualified worker*. Assume, for now, that all the workers have the same fixed lifetime T; they are all qualified workers, and all workers get paid the same amount, r per task.[6]

We denote the marginal pdf of reservation wage by $f_W(w)$ and the cdf of the same by $F_W(w)$. Since the client pays S for each successfully completed task, each task submitted by a qualified worker is worth S to the platform, minus the cost of labor. When the offered price is r, the net profit

[4] The quality of a worker varies depending on how much time the worker invests for each particular task. The longer a worker spends on a task, the higher the quality of the outcome is. Essentially, there is a tradeoff between the quality and the productivity of a particular worker. We model each worker as a profit-maximizer: given the payment scheme $R(q)$ published by the platform, the worker computes the expected productivity $N(q')$ (i.e., number of tasks the worker can complete within a time unit, with quality q') for all different levels of quality q' and then sets its own quality q^* by choosing a quality value that maximizes his expected profits (i.e., $q^* = \arg\max_q N(q)R(q)$). Selecting q^* also defines the lifetime $t^{(k)}$ of the worker when the worker has a fixed amount of time available to work.

[5] The function $g(\mathbf{e}^{(k)})$ maps the confusion matrix into a scalar value. We will discuss the specifics later in the paper.

[6] We will discuss in Section 7.1 how we can "convert" unqualified workers into qualified equivalents using redundancy, and measure how many unqualified workers are needed to create one qualified "aggregate" worker.

from a qualified worker with reservation wage w is as follows (assume $0 \leq r \leq S$):

$$\mathit{Profit}(r, w) = \begin{cases} 0 & : w > r \\ (S - r) \cdot T & : w \leq r \end{cases} \tag{1}$$

Therefore, the expected net profit from a worker is:

$$\mathit{Profit}(r) = \int_0^{\infty} \mathit{Profit}(r, w) \cdot f_W(w) \mathrm{d}w = \int_0^{r} (S - r) \cdot T \cdot f_W(w) \mathrm{d}w = F_W(r) \cdot (S - r) \cdot T \tag{2}$$

The optimal price r^* is given by the solution to the maximization problem:

$$r^* = \arg\max_r \mathit{Profit}(r) = \arg\max_r F_W(r) \cdot (S - r) \cdot T \tag{3}$$

Taking the derivative of $F_W(r) \cdot (S - r) \cdot T$ with respect to r and setting it to zero, we get

$$r^* = S - \frac{F_W(r^*)}{f_W(r^*)} \tag{4}$$

Example 1 *The client is willing to pay $S = 1$ for each successfully completed example. The distribution of the reservation wage w follows a uniform distribution on the interval $[0, 1]$. In this case, we have:*

$$f_W(r) = \begin{cases} 1 & : r \in [0, 1] \\ 0 & : \mathit{otherwise} \end{cases} \quad ; \quad F_W(r) = \begin{cases} 0 & : r < 0 \\ r & : r \in [0, 1] \\ 1 & : r > 1 \end{cases}$$

Putting the values in Equation 3, we get $r^* = 0.5$. In a slightly more general case, if the payment is S, and the reservation wage w follows a uniform distribution on the interval $[l, h]$, we get that:

$$f_W(r) = \begin{cases} \frac{1}{h-l} & : r \in [l, h] \\ 0 & : \mathit{otherwise} \end{cases} \quad ; \quad F_W(r) = \begin{cases} 0 & : r < l \\ \frac{r-l}{h-l} & : r \in [l, h] \\ 1 & : r > h \end{cases}$$

In that case, $r^* = (S + l)/2$, when $S \in [l, 2h - l]$. If $S < l$, there is no worker engagement, and if $S > 2h - l$ then $r^* = h$.

5 Quality Estimation

In the previous section, we derived a simple pricing policy, assuming that all workers generate work of acceptable quality, or that we can separate the qualified from the "unqualified" workers. In reality, though, worker quality is typically unobservable to the platform and we need to estimate the quality of workers through testing. Towards this, in Section 5.1, we describe a scheme that uses redundancy, together with optional testing, to generate estimates about the type and prevalence of the errors committed by the workers in their tasks. Then, in Section 5.2, we investigate some problems of the error rate as a measure of quality, and describe how to generate an unbiased quality estimator, using a decision theoretic framework.

5.1 Worker Quality Estimation

5.1.1 Expectation Maximization for Error Rate Estimation:

An early paper by Dawid and Skene (1979) described how we can estimate the error rates of workers that perform a task, when we do not have access to the correct outcomes but can only observe the worker output. The particular application examined in Dawid and Skene (1979) was the estimation of diagnostic error rates when doctors examine patients, but there is no known correct answer for the diagnosis. The basic idea is to rely on redundancy, i.e., to obtain multiple opinions about the diagnosis. The algorithm iterates until convergence, following two steps: (1) estimates the true response for each patient, using observations given by the doctors, accounting for the error-rates of each doctor; and (2) estimates the error-rates of doctors by comparing the submitted observations to estimated correct diagnosis. The final output of this expectation-maximization algorithm is the estimated diagnosis for each patient and the estimated error-rate represented by "confusion matrix" for each doctor. We rephrase the algorithm into our problem setting, in which workers assign labels to objects.

Input: The input of the algorithm is a set of labels provided by workers. We use O to refer to all the objects, and $l_{(o)}^{(k)}$ to refer to the label that worker k assigns to object o. For convenience, notation $K^{(o)}$ is used to denote the set of workers that assign labels to object o, and notation $O^{(k)}$ is used to denote the set of objects labeled by worker k.

Output: The output of the algorithm is the confusion matrix $\mathbf{e}^{(k)}$ for each worker (k), the prior $\boldsymbol{\pi}$ for each class (π_i represents the prior for class i), and the class probability estimation $\mathbf{p}^{(o)}$ for each object (o) ($p_i^{(o)}$ represents the estimated probability that the true label of object (o) is i).

Initialization: For each object (o), the initial probability estimates of the true class are:[7]

$$p_i^{(o)} = \frac{\sum_{(k)\in K^{(o)}} I(l_{(o)}^{(k)} = i)}{\left|K^{(o)}\right|} \tag{5}$$

Error rate estimation: For each worker (k), given the class probability estimation for each object in $O^{(k)}$, the maximum likelihood estimates of the error rates are:

$$e_{ij}^{(k)} = \frac{\sum_{(o)\in O^{(k)}} p_i^{(o)} I(l_{(o)}^{(k)} = j)}{\sum_q \sum_{(o)\in O^{(k)}} p_i^{(o)} I(l_{(o)}^{(k)} = q)} \tag{6}$$

One differentiating factor of our work, is that the $p_i^{(o)}$ estimates used in the equation above do *not* include the labels of worker (k). This allows us to estimate the quality of the worker, based purely on comparing the outcome of a worker against *other* workers, as opposed to comparing a worker against itself.

Class prior estimation: The prior for each class i is estimated as:

$$\pi_i = \frac{\sum_{(o)} p_i^{(o)}}{|O|} \tag{7}$$

Class probability estimation: When the individual confusion matrix $\mathbf{e}^{(k)}$ and the class priors $\boldsymbol{\pi}$ are known, we can apply Bayes' Theorem to obtain probability estimates of true labels for each object (o)[8].

$$p_i^{(o)} \propto \pi_i \prod_{(k)\in K^{(o)}} \prod_m (e_{im}^{(k)})^{I(l_{(o)}^{(k)}=m)}$$

where all terms not involving i are absorbed into the constant of proportionality. Thus

$$p_i^{(o)} = \frac{\pi_i \prod_{(k)\in K^{(o)}} \prod_m (e_{im}^{(k)})^{I(l_{(o)}^{(k)}=m)}}{\sum_q \pi_q \prod_{(k)\in K^{(o)}} \prod_m (e_{qm}^{(k)})^{I(l_{(o)}^{(k)}=m)}} \tag{8}$$

Algorithm 1 presents a sketch of the process. The algorithm iterates between estimating the class probability distribution $\mathbf{p}^{(o)}$ for each object, and estimating the confusion matrix $\mathbf{e}^{(k)}$ for each worker.

[7] The initial probability estimates are based on the fraction of labels assigned by workers in each class.

[8] We assume that the labels assigned by workers for any object (o) are independent, given the true class.

```
Input: Set of Labels $\{l_{(o)}^{(k)}\}$
Output: Confusion matrix $\mathbf{e}^{(k)}$ for each worker $(k)$, Class priors $\boldsymbol{\pi}$, Class probability estimates $\mathbf{p}^{(o)}$
        for each object $(o)$
Using Eq. 5, initialize class probability estimates $\mathbf{p}^{(o)}$ for each object $(o)$;
while not converged do
    Using Eq. 6, estimate the confusion matrix $\mathbf{e}^{(k)}$ for each worker $(k)$;
    Using Eq. 7, estimate the class priors $\boldsymbol{\pi}$;
    Using Eq. 8, compute the class probability estimates $\mathbf{p}^{(o)}$ for each object $(o)$;
end
return $\{\mathbf{e}^{(k)}\}$, class priors $\boldsymbol{\pi}$, $\{\mathbf{p}^{(o)}\}$
```

Algorithm 1: The expectation maximization algorithm for estimating error rates of workers.

5.1.2 From MLE to Bayesian Estimation:

The expectation maximization algorithm performs well when we have significant number of observations completed by each worker. Unfortunately, participation in crowdsourcing environments follows a very skewed distribution (Stewart et al., 2010; Nov et al., 2011) with only a few workers contributing a lot, while the majority submit only a few tasks. In such a setting, maximum likelihood approaches result in overly confident estimates of the error rates and the quality of the workers. Consider the following example:

Example 2 *There are two workers that work on a labeling problem. The history of worker A and worker B are:*

$$\mathbf{n}^{(A)} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad ; \quad \mathbf{n}^{(B)} = \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$$

where $n_{ij}^{(k)}$ is the number of times that worker (k), when presented with an object of true class i, classifies the object into category j. It is easy to find that the two workers have exactly the same confusion matrix, when expressed in terms of observed error rates. However, we do not have the same level of confidence in the two estimates: Worker B is more likely to be a perfect worker, while our assessment for worker A is much more uncertain.

Therefore, we move from maximum likelihood estimates to Bayesian ones. If the true class of an example is i, we model the error rates of the worker as a Dirichlet distribution with parameter the vector $\boldsymbol{\theta}_i$. The values in $\boldsymbol{\theta}_i$ are based on the number of times that the worker classified objects of class i into class j ($\theta_{ij} = n_{ij} + 1$, if we start with a uniform distribution as a prior for the error rates of the worker). Following this strategy, the error rates of a worker can be fully captured by a set of

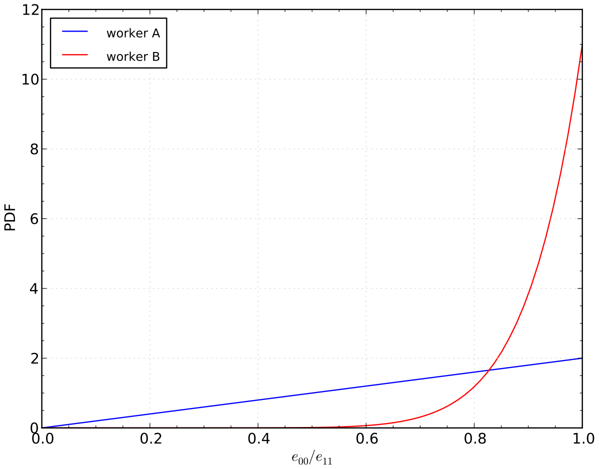


Figure 2: Probability distribution for e_{00}/e_{11}

Dirichlet distributions (which reduce to Beta distributions for the binary case).

Example 3 *Let us revisit the two workers from Example 2. Following the Bayesian approach,*[9] *we have:* $e_{00}^{(A)}/e_{11}^{(A)} \sim Beta(2,1)$ *and* $e_{00}^{(B)}/e_{11}^{(B)} \sim Beta(101,1)$*. Figure 2 shows the respective probability density functions (pdf) for the diagonal elements in the confusion matrices of the two workers. The uncertainty about the worker A is higher, compared to the estimated distribution for worker B.*

All the procedures in Algorithm 1 remain the same, except for the error rate estimation. We now need to estimate $\boldsymbol{\theta}^{(k)}$, which are given by

$$\theta_{ij}^{(k)} = \alpha_{ij}^{(k)} + n_{ij}^{(k)} = \alpha_{ij}^{(k)} + \sum\nolimits_{(o)\in O^{(k)}} p_i^{(o)} I(l_{(o)}^{(k)} = j) \tag{9}$$

where $\boldsymbol{\alpha}^{(k)}$ captures the prior we impose on the error rates of worker (k).[10] This Bayesian approach yields a full distribution of confusion matrix for a worker, and ideally we would want to compute $\mathbf{p}^{(o)}$ using a weighted integration of its value under all different realizations of $\mathbf{e}^{(k)}$. Since it is very computationally expensive, currently we use the mean of the Dirichlet distribution as error

[9] Assume we have as prior the uniform distribution, $Beta(1,1)$

[10] Since crowdsourcing workers tend to have heterogeneous levels of quality, we use uninformative priors in our estimation: i.e., $\alpha_{ij}^{(k)}$=1 $\forall i,j$.

```
Input: Set of Labels $\{l_{(o)}^{(k)}\}$
Output: Confusion matrix $\mathbf{e}^{(k)}$ for each worker $(k)$, Class priors $\boldsymbol{\pi}$, Class probability estimates $\mathbf{p}^{(o)}$
        for each object $(o)$
Using Eq. 5, initialize class probability estimation $\mathbf{p}^{(o)}$ for each object $(o)$;
while not converged do
    Using Eq. 9, estimate the Dirichlet parameter matrix $\boldsymbol{\theta}^{(k)}$ for each worker $(k)$;
    Using Eq. 10, estimate the confusion matrix $\mathbf{e}^{(k)}$ by applying the mean of the Dirichlet distribution;
    Using Eq. 7, estimate the class priors $\boldsymbol{\pi}$;
    Using Eq. 8, compute the class probability estimates $\mathbf{p}^{(o)}$ for each object $(o)$;
end
return $\{\mathbf{e}^{(k)}\}$, class priors $\boldsymbol{\pi}$, $\{\mathbf{p}^{(o)}\}$
```

Algorithm 2: The Bayesian version of EM algorithm for worker quality estimation.

rates when computing the class probability estimates for objects: [11]

$$e_{ij}^{(k)} = (\alpha_{ij}^{(k)} + n_{ij}^{(k)}) / \sum_{m=1}^{L} (\alpha_{im}^{(k)} + n_{im}^{(k)}) \tag{10}$$

We adapt the expectation maximization approach as shown in Algorithm 2.

One virtue of the algorithm is that it can seamlessly integrate redundancy and testing. If the platform has access to "gold" data (i.e., objects for which the correct answers are already known), then these objects can be used to speed up the worker quality estimation process. The platform can simply insert a few gold data points in the stream of tasks completed by each worker, and ask workers to provide answers. To handle such gold data the algorithm would be modified to skip updating the true class of the "gold" objects, in the step of class probability estimates in Algorithm 1 and 2, but rather keep the class estimates fixed into correct ones.

5.2 Generating Unbiased Quality Measurements

The confusion matrix $\mathbf{e}^{(k)}$ for each worker (k) alone cannot provide sufficient information when our objective is to assess the value of the workers. A naive method is to simply sum up the non-diagonal entries of the matrix $\mathbf{e}^{(k)}$, weighting each error rate by the estimated prior of each class. Unfortunately, this approach would wrongly reject biased but careful workers. Consider the following example:

Example 4 *Consider two workers that label web sites into two classes:* porn *and* notporn*. Worker A is* always *incorrect: labels all* porn *web sites as* notporn *and vice versa. Worker B classifies all*

[11] More sophisticated techniques could be employed instead.

web sites, irrespectively of their true class, as porn. *Which of the two workers is better? A simple error analysis indicates that the error rate of worker A is 100%, while the error rate of worker B is "only" 50%.*[12] *However, it is not hard to see that the errors of worker A are easily reversible, while the errors of worker B are irreversible. In fact, worker A is a perfect worker, while worker B is a spammer.*

As a more realistic problematic case, consider the same problem as before, but with a skewed class distribution: 95% of the web sites fall in the category of notporn, *while only 5% of the web sites are* porn. *A strategic spammer C classifies all web sites, irrespectively of their true class, as* notporn. *By doing so, he can achieve a low error rate of 5%, and may well be considered as a low-error, and thus high-quality, worker if we employ the simple error analysis.*

So, naturally a question arises: Given accurate estimates of the confusion matrix $\mathbf{e}^{(k)}$ for each worker (k), how can we separate low-quality workers from high-quality, but biased, workers? How can we identify the strategic spammers? How can we separate systematic biases from the intrinsic, non-recoverable error rates? We examine these questions next.

We start with the following observation: Each worker assigns a "hard" label to each object. Using the error rates for this worker, we can transform this assigned label into a "soft" label (i.e., posterior estimate), which is the best possible estimate that we have for the *true* class. So, if we have L possible classes and the worker assigns class j as a label to an object, we can transform this "hard" assigned label into the "soft" label:

$$\left\langle \pi_1 \cdot e_{1j}^{(k)}, \ldots, \pi_L \cdot e_{Lj}^{(k)} \right\rangle \tag{11}$$

where π_i is the prior that the object will belong to class i and $e_{ij}^{(k)}$ is the probability that worker (k) classifies into class j an object that in reality belongs to class i. We should note that the quantities above need to be normalized by dividing them with

$$\pi_j^{(k)} = \sum_{i=1}^{L} \pi_i \cdot e_{ij}^{(k)} \tag{12}$$

the probability that worker (k) assigns label j to any object.

Example 5 *Take the case of worker A from Example 4. When this worker assigns a label of* Porn

[12] Assume, for simplicity, equal priors for the two classes.

(assume that porn is class 1), then the corresponding soft label has all the "probability mass" in the NotPorn *category:*

$$\underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\textit{Assigned: Porn}} \quad \Rightarrow \quad \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\textit{Corrected to: NotPorn}}$$

On the contrary, for worker B, who always assigns porn, *the corresponding corrected soft label does not give us any information; the soft label simply says that the best guess are simply the class priors:*

$$\underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{\textit{Assigned: Porn}} \quad \Rightarrow \quad \underbrace{\begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix}}_{\textit{Corrected to: Class priors}}$$

So, what can we do with these soft labels? The key idea is to estimate the *expected cost of each (soft) label.* To estimate the cost of a soft label, we need to consider the costs associated with all possible classification errors. In the simplest case, we have a cost of 1 when an object is misclassified, and 0 otherwise. In a more general case, we have a cost c_{ij} when an object of class i is classified into category j.

Proposition 6 *Given the classification costs* $\mathbf{c}$ *and a soft label* $\mathbf{p} = \langle p_1, p_2, \ldots, p_L \rangle$*, the expected cost of the soft label* $\mathbf{p}$ *is:*

$$ExpCost\left(\mathbf{p}\right) = \min_{1 \leq j \leq L} \sum_{i=1}^{L} p_i \cdot c_{ij} \tag{13}$$

The proof is rather simple. The expected classification cost if we report j as the true class is equal to the posterior probability of the object belonging to class i (which is p_i), multiplied with the associated cost of classifying an object of class i into class i (which is c_{ij}). The Bayesian decision is to report the category j with the minimum expected classification cost across all classes. The expected cost can help us make the best classification decision in the case where we receive only a single label per object.

It turns out that workers with confusion matrices that generate posterior labels with probability mass concentrated into a single class (i.e., confident posterior labels) will tend to have low estimated cost, as the minimum sum in Equation 13 will be close to 0. On the contrary, workers that tend to generate posterior labels with probabilities spread out across classes (i.e., uncertain posterior labels) will tend to have high misclassification costs.

```
Input: Confusion matrix $\mathbf{e}$, Misclassification cost matrix $\mathbf{c}$, Class prior vector $\boldsymbol{\pi}$
Output: Expected cost $cost^{(k)}$ for each worker $(k)$
foreach worker $(k)$ do
    Estimate $\pi_l^{(k)}$ (how often the worker $(k)$ assigns label $l$), using Eq. 12;
    $cost^{(k)} = 0$;
    foreach label $l$, assigned with probability $\pi_l^{(k)}$ do
        Using Eq. 11, compute the posterior probability $\mathbf{soft}^{(k)}(l)$ that corresponds to label $l$ assigned
        by worker $(k)$;
        Using Eq. 13, compute $Cost(\mathbf{soft}^{(k)}(l))$ for the soft label;
        $cost^{(k)}$ += $Cost(\mathbf{soft}^{(k)}(l)) \cdot \pi_l^{(k)}$;
    end
end
return $cost^{(k)}$ for each worker $(k)$
```

Algorithm 3: Estimating the Expected Cost of each Worker

Example 7 *Consider the costs for the workers A and B from the previous examples. Assuming equal priors across classes, and $c_{ij} = 1$, if $i \neq j$ and $c_{ij} = 0$, if $i = j$, we have the following: The cost of worker A is 0, as the soft labels generated by A are $\langle 0, 1 \rangle$ and $\langle 1, 0 \rangle$. For worker B, the cost is 0.5 (the maximum possible) as the soft labels generated by B are all $\langle 0.5, 0.5 \rangle$ (i.e., highly uncertain).*

Given that we know how to compute the expected cost for each label, we can now easily estimate the expected cost for each worker (k). We first compute the priors $\pi_j^{(k)}$ (see Equation 12), which are the prior probabilities of the worker assigning each label j to an object. Then we compute the posterior label vector that corresponds to the assigned label (see Equation 11). Given the posterior label vector, we use Equation 13 to compute the expected cost of each assigned label. Now, knowing how often the worker assigns a label and the expected cost, we can compute the average expected cost of each worker. Algorithm 3 illustrates the process.

As expected, perfect workers will have a cost of zero and random workers or spammers will have high expected costs. Notice, as illustrated in the example above, that it is not necessary for a worker to return the correct answers in order to have low costs! As long as the errors are predictable and reversible, the worker is assigned a low expected cost. Effectively, *systematic, and hence reversible, biases are corrected and not taken into consideration when evaluating the quality of a worker.*

Our quality metric based on expected misclassification costs resolves quite a few issues with online workers who exhibit systematic biases in their answers but who also put a lot of effort in coming up with the answers. Prior approaches that relied on agreement generate a significant number of rejections for such workers, which in turn alienates such high-quality workers, and discourages them from working with employers that rely on agreement.

6 Dynamic Resource Allocation: Labeling Data and Learning the Workers

In the previous section, we have focused on a static setting: we have all the data, we perform the analysis, and then examine data and worker quality. In reality, labels are often obtained incrementally and dynamically: either we have workers that arrive and need to get assigned to label specific objects or we have objects that arrive and need to be allocated to workers. The allocation of resources faces an exploration-exploitation tradeoff. We can try to purely "exploit": *label examples* as well as possible, ignoring the objective of learning the worker quality and let the worker estimation be a side-effect. Or we can "explore" and try to *learn the worker quality*, allowing us to be more confident for the quality of our labels in the future.

In Section 6.1, we focus on the *data labeling* scenario, in which the ultimate goal is to ensure data quality. This aligns with the goal of the platform in our current model—promising a certain level of data quality to clients. In Section 6.2, we discuss the *worker quality learning* scenario, where the aim is to accurately learn the quality of workers.

6.1 Ensuring Data Quality

The price that clients offer to the platform is contingent on the assurance of a certain level of data quality. Therefore, it is important to be able to monitor efficiently the quality level of the delivered data, and to allocate worker resources appropriately, to achieve this goal. The key insight is that, given the set of workers that have labeled an object, we can both estimate the most likely correct label for the labeled object *and* we can estimate what is the probability of this label being incorrect (Section 6.1.1). Furthermore, given the misclassification costs, we can estimate the expected misclassification cost for the object, and allocate labeling resources in a way that increases data quality in a cost-effective manner (Section 6.1.2).

6.1.1 Estimating the quality of a single data point

In the labeling process, we ask workers to label different objects. Since the platform is trying to allocate resources optimally, the key question is how many workers we should assign to label each object? The confidence about the classification decision of each object increases, in expectation, as we get more workers to inspect and label it. Therefore, the more workers we assign to each object, the higher the labeling quality. At the same time, we want to minimize the labor costs for the platform. Since the goal is to have an overall data quality higher than the quality promised in the SLA, it is optimal to assign to each object just enough labels so that the error rate in the labeled object is lower than the one promised in the SLA.

How can we estimate the quality of the labeling? Assume that we have an object that has been labeled by m workers ($m \geq 2$), and that these workers assigned a multiset of labels[13] $\mathbf{j} = \{j_1, j_2, \cdots, j_m\}$, where j_s is the label assigned by the s-th worker in the set. Using the results from Section 5, we assume that we have an estimate of the confusion matrix for each worker (k), which we denote as $\mathbf{e}^{(k)}$ (see Section 4). We start by asking: given an object of class l, what is the probability of seeing a particular label assignment $\mathbf{j} = \{j_1, j_2, \cdots, j_m\}$?

$$P(\mathbf{j}|e_{ij}^{(1)}, e_{ij}^{(2)}, \cdots, e_{ij}^{(m)}, l) = \prod_{s=1}^{m} e_{lj_s}^{(k)} \tag{14}$$

Simply using Bayes' Rule, we have the posterior probability of the object belonging to class l, given by:

$$\begin{aligned} P(l|\mathbf{j}, e_{ij}^{(1)}, e_{ij}^{(2)}, \cdots, e_{ij}^{(m)}) &\propto \pi_l \cdot P(\mathbf{j}|e_{ij}^{(1)}, e_{ij}^{(2)}, \cdots, e_{ij}^{(m)}, l) \\ &= \pi_l \cdot \prod_{s=1}^{m} e_{lj_s}^{(k)} \end{aligned} \tag{15}$$

Therefore, after we observe $\mathbf{j}$ as the assigned label set, the "soft" label is:

$$\left\langle \pi_1 \cdot \prod_{k=1}^{m} e_{1j_k}^{(k)}, \ldots, \pi_L \cdot \prod_{k=1}^{m} e_{Lj_k}^{(k)} \right\rangle \tag{16}$$

[13] Note that the assigned labels are conditionally independent, given the true class.

Same as before, we need to normalize the quantities above by dividing them with

$$\sum_{l=1}^{L} \pi_l \cdot \prod_{s=1}^{m} e_{lj_s}^{(k)} \tag{17}$$

Using Proposition 6 and the corresponding Equation 13 from Section 5.2, we can estimate what is the expected misclassification cost for this object. Intuitively, objects that have a posterior probability assigned only to one class, will have low expected misclassification cost. On the other hand, objects with a posterior probability that is spread across classes with high misclassification cost are deemed as having low quality, and therefore need higher level of (additional) attention.

6.1.2 Selective Labeling Based on Expected Misclassification Cost

Given the quality estimate (i.e., expected classification cost) for each object, we can devise a labeling policy that assigns workers to objects. A natural policy is to focus on the examples with the *highest expected classification cost.* When a worker arrives, the worker is assigned to label the object with the highest expected cost, as long as the object has an expected cost higher than the one promised in the SLA[14]. (If the cost is lower, the object is ready to be delivered to the client.)

Two minor points might limit the applicability of the labeling strategy described above in real-world large data environments. First, at each time point, we need to compute the expected classification cost for all the objects and choose the one with the highest cost, which is computationally expensive. Second, we tend to assign workers to objects for which we are less certain about first; however, an accurate estimation of worker quality relies on a good estimation of the labels for the objects that the worker has worked on. This poses a disadvantage for the early-coming workers since they need to wait for a long time to get their expected cost correctly estimated. To avoid the computational complexity and latency in worker quality updates, we divide the full set of objects into a number of subsets $N = \{N_1, N_2, \cdots, N_n\}$ where each N_i only contains a relatively small number of objects. We will first focus on the subset N_1, and then N_2, and so on.

As we are going to demonstrate in our experimental results (Section 8.2), this strategy improves upon the current state-of-the-art strategy, the *NLU* strategy from Ipeirotis et al. (2013). The advantage of this policy is that we can prioritize the examples without knowing anything about the worker pool. A potential improvement, if we know the confusion matrix $\mathbf{e}$ of the worker, is to

[14] In the actual implementation, we can either meet the SLA in expectation, or with a certain confidence level. A higher confidence level reduces the risk of failing to meet the standard but demands more labeling resources.

compute the *expected marginal improvement in classification cost*, for each object, by comparing the current classification cost with the expected future classification cost.[15] Unfortunately, this approach is computationally inefficient, as we need to compute on the fly the marginal improvement for a large number of objects, and also in practice it does not offer any significant improvements compared to the worker-agnostic strategy described above: The objects that are typically picked under this strategy are very often the objects with the currently highest expected classification cost.

6.2 Learning Worker Quality

In Section 6.1, we described how to prioritize objects that need more labels. In some cases, the primary objective is to know better the quality of workers (Bachrach et al., 2012). In contrast to the case of labeling objects, even when a worker labels a very large number of objects, we will never reach the point of "zero expected cost." We will simply estimate very accurately the mean classification cost incurred when this worker labels an object. Naturally, we can aim at minimizing the variance of the cost estimation, instead. However, the question arises: What is the value of minimizing the uncertainty? For the objects, we are trying to bring the expected cost below a specific threshold τ_c. For workers, there is no clear objective.

The key here is how to define the *cost of uncertainty.* Following the quality estimation method in Section 5, at each time point, we have a full distribution of the potential confusion matrix for the worker.[16] As we describe in Section 7, each confusion matrix can be mapped into a payment value, allowing us to infer a distribution of potential payments for the worker. Given that the payment for the worker needs to be a deterministic value, we need to collapse the distribution into a single value. The uncertainty in this context shows the potential for underpayment and overpayment of the worker.

In this setting, the cost of positive uncertainty can be defined as either the total uncertainty about the offered wage (integration over all over- and under-payments), or the potential for overpayment (integration over over-payments). This cost becomes the priority value for the different workers when allocating labels. In the next section, we describe how to match different worker quality levels in payments, allowing the implementation of such priority schemes.[17]

[15] Given the current posterior of the object, we can estimate the probability that the worker assigns different labels to the object, and then use Equations 15 and 13 to estimate the future classification costs.

[16] While this allows us to get the resulting distribution of expected cost for a worker, we do not have a clear objective on how to value the uncertainty about the misclassification cost.

[17] For brevity and focus, we do not investigate further the direction of deciding how to test the workers. There is

7 Quality-sensitive Pricing

As we discussed in Section 4.2, the pricing for qualified workers, i.e., individual workers with labeling quality above the one promised in the SLA, is relatively straightforward. However, many workers in crowdsourcing markets do not have high enough quality to satisfy the SLA promised by the platform. In fact, there may be cases where *no* worker satisfies the desired quality.[18] While we can consider such "unqualified" workers to be ineligible to work on the tasks (i.e., office a price of zero for their work), this is very inefficient. Multiple papers in the literature Sheng et al. (2008); Snow et al. (2008); Welinder et al. (2010); Raykar et al. (2010); Ipeirotis et al. (2010); Bachrach et al. (2012) show that using multiple, low-quality workers can be used to generate results that have high quality. The focus of this section is to examine how to reward such "unqualified" workers.

7.1 Equivalence of Unqualified and Qualified Workers

The objective of the platform is to get data labeled with classification cost lower than the level determined by the SLA. In Section 4.2, we described how we price the submitted work of qualified workers, that provide labels with low expected cost. In Section 6.1, we also described how we can improve data quality (and decrease expected cost for an object), by allocating multiple workers to label it. Therefore, a *set of unqualified workers that in tandem can generate labels of high quality* can be considered equivalent to a single, qualified worker. So, in our work, we propose to pay unqualified workers according to the level of redundancy that is required to reach the required quality level.

Example 8 *Suppose that a client has a binary classification problem with equal priors, misclassification costs set to* 1*, and an SLA that requires a classification cost lower than* 0.1*. If we have workers with a confusion matrix of* $\mathbf{e} = \begin{pmatrix} q & 1-q \\ 1-q & q \end{pmatrix}$*, how many workers do we need to assign to each object, to achieve the SLA requirement? Figure 3 shows the relationship between the number of workers and the integrated expected cost with the value of* q *ranging from* 0.60 *to* 0.90*. The black dash line indicates the SLA-promised cost level. We can see that:*

1. *A worker with* $q = 0.9$ *is a qualified worker, and should get a wage following the pricing model of Section 4.2.*

already significant related literature in that front in the field of psychometrics and in the field of acceptance sampling in operations.

[18] Or, more commonly, it is not cost-efficient to allow workers to be slow and careful in order to meet the SLA requirements.

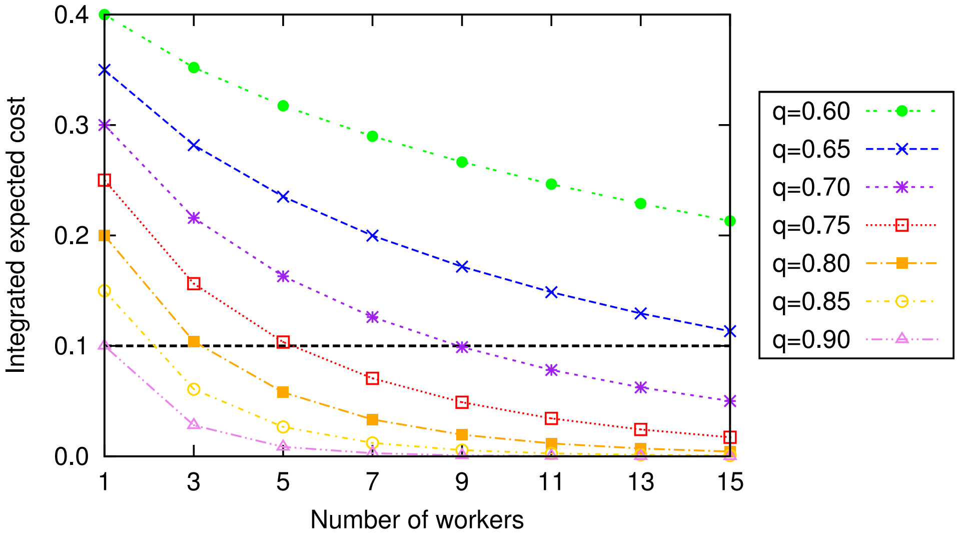


Figure 3: The relationship between the number of workers and integrated expected cost

2. *A worker with $q = 0.8$ is unqualified. However, a set of 3 workers with $q = 0.8$ generate labeling of SLA quality. Therefore a worker with $q = 0.8$ should receive $1/3$ of the wage of a qualified worker.*

3. *We need 9 workers with $q = 0.7$ to reach the SLA quality, therefore a worker with $q = 0.7$ should receive $1/9$ of the wage of a qualified worker.*

□

The example above illustrates that *value* of a worker is inversely proportional the number of workers with the same error rates required to achieve the "accept" level of cost. The example illustrates the process for a worker with a specific confusion matrix; next, we show the process for estimating the value of a worker with an arbitrary confusion matrix $\mathbf{e}$.

Definition 9 *The* value $v(\mathbf{e})$ *of a worker with a confusion matrix* $\mathbf{e}$ *is:*

$$v(\mathbf{e}) = \frac{S}{d(\mathbf{e})} \tag{18}$$

where $d(\mathbf{e})$ is the number of workers with confusion matrix $\mathbf{e}$ required to reach the SLA-defined classification cost of τ_c, and S is the price charged to a client for a unit of SLA-compliant work. For qualified workers $d(\mathbf{e}) = 1$, while for unqualified workers $d(\mathbf{e}) > 1$.

The key challenge is to estimate the value $d(\mathbf{e})$ for an arbitrary confusion matrix $\mathbf{e}$. For this, we need to estimate the number of workers with identical confusion matrix $\mathbf{e}$ that are required to generate labeling of acceptable quality. Unfortunately, except for very simple cases, there is no closed form solution to this problem, and the computational complexity increases exponentially with the value of $d(\mathbf{e})$. Hence, we resort to a Monte Carlo approach for estimating $d(\mathbf{e})$.

The approach works as follows. We assume that we have m workers with identical confusion matrix $\mathbf{e}$ who assign labels to an object. This generates a label assignment $\mathbf{l} = \{l_1, \cdots, l_m\}$, which, due to the exchangeability of the labels, can be represented as a count of the different class labels $\mathbf{n} = \{n_1, \cdots, n_L\}$[19]. When the true class label is i (which occurs with probability π_i), this label assignment happens with probability $\mathit{Mult}(\mathbf{n}|m, \mathbf{e}_{i\cdot}) = \binom{m}{n_1, \cdots, n_L} \cdot \prod_{j=1}^{L}(e_{ij})^{n_j}$, which is the probability mass function (pmf) of the multinomial distribution with parameters m (count of trials) and $\mathbf{e}_{i\cdot}$ (the line of the confusion matrix $\mathbf{e}$ that corresponds to the class i). Integrating this over all the classes, we get the overall probability of seeing $\mathbf{n}$ is:

$$P(\mathbf{n}) = \sum_{i=1}^{L} \pi_i \cdot \mathit{Mult}(\mathbf{n}|m, \mathbf{e}_{i\cdot}) = \binom{m}{n_1, \cdots, n_L} \sum_{i=1}^{L} \pi_i \cdot \prod_{j=1}^{L}(e_{ij})^{n_j} \tag{19}$$

Following the same procedure in Section 6.1.1, for each label assignment $\mathbf{n} = \{n_1, \cdots, n_L\}$, the “soft” label before normalization is proportional to:

$$\left\langle \pi_1 \cdot \prod_{j=1}^{L}(e_{1j})^{n_j}, \ldots, \pi_L \cdot \prod_{j=1}^{L}(e_{Lj})^{n_j} \right\rangle \tag{20}$$

The expected misclassification cost associated with the label assignment $\mathbf{n}$ is then estimated using Equation 13. By repeating the process multiple times (across different label assignments, using Monte Carlo sampling), we get the average misclassification cost when using m workers with confusion matrix $\mathbf{e}$. Knowing how to compute the integrated expected cost, the utility derivation becomes easier. Given a worker with specific confusion matrix $\mathbf{e}$, we simply find the number of workers $d(\mathbf{e})$ we need to achieve the required cost level.

Note though, that $d(\mathbf{e})$ is not always an integer, and the process described above returns the smallest integer value that satisfies the cost requirements. For example, in Figure 3, one worker with $q = 0.85$ has an expected cost of 0.15 ($> \tau_c = 0.10$), the smallest integer value of $d(\mathbf{e})$ that achieves

[19] $\sum_{j=1}^{L} n_j = m$ holds naturally.

```
Input: Confusion matrix $\mathbf{e}$, Misclassification cost matrix $\mathbf{c}$, Class prior vector $\boldsymbol{\pi}$
Output: Value $v(\mathbf{e})$
$m = 1$;
while $m \leq D$ do
    $SumCost = 0$;
    $cnt = 0$;
    while cost not converged do
        Pick an object class $i$, with probability proportional to the prior $\pi_i$;
        Draw an label assignment $\mathbf{n}$ from the multinomial distribution with $m$ trials and probability
        parameters $\mathbf{e}_{i\cdot}$;
        Using Eq. 20, compute the posterior probability vector $\mathbf{soft}(\mathbf{n})$ that corresponds to $\mathbf{n}$;
        Using Eq. 13, compute $Cost(\mathbf{soft}(\mathbf{n}))$ for the posterior probability vector ;
        $SumCost \mathrel{+}= Cost(\mathbf{soft}(\mathbf{n}))$;
        $cnt++$;
    end
    $cost = \frac{SumCost}{cnt}$;
    if $cost \leq \tau_c$ then
        break;
    end
    $m = m + 1$;
end
Compute $d(\mathbf{e})$ using logarithmic interpolation or logarithmic regression ;
Using Eq. 18, compute the value $v(\mathbf{e})$ of the worker;
return $v(\mathbf{e})$
```

Algorithm 4: Estimating the value $v(\mathbf{e})$ of a worker with confusion matrix $\mathbf{e}$

a cost below the threshold $\tau_c = 0.10$ is three workers; with three workers, though, we have a cost of 0.06, which is substantially lower than the target value. For this case, we employ *logarithmic interpolation* between these two points to get the approximate number. The interpolation can give us an approximate value if the left-end point and right-end point are given. But when the worker quality is low, it becomes computationally expensive to get the reference point (e.g., $q = 0.60$ in Figure 3). To predict data points outside the computation limit[20] (i.e. $d(\mathbf{e})$ is too large), we apply *logarithmic regression*.[21] Algorithm 4 illustrates the overall process.

7.2 Optimal Pricing Mechanism for Workers with Heterogeneous Quality

So far, we know how to compute the value of all workers, even in the presence of heterogeneous quality levels: the value of each qualified worker is S, and the value of each unqualified worker is $S/d(\mathbf{e})$, following the analysis in Section 7.1. We now expand the simple pricing model of

[20] We denote the computation limit as D, which represents for the maximum number of workers we use for computing expected cost.

[21] Both extrapolation and regression can be used to predict data outside the range. Extrapolation is not stable since it only includes the last two points; however, regression including all the points might not be able to reflect the newest trend. In our paper, we run logarithmic regression on the last 4 points.

Section 4.2, alleviating the assumption that all workers are qualified and produce work above the quality requirements of the SLA. The core idea is that instead of treating each worker equally, we effectively "replace" $d(\mathbf{e})$ unqualified workers in the pool with a single qualified worker.

We use $f(\mathbf{e}, w, t)$ to denote the joint distribution of confusion matrix, reservation wage, and lifetime of the workers in the crowdsourcing pool. For a given worker with confusion matrix $\mathbf{e}$, we first compute the value of this worker $v(\mathbf{e})$ using Algorithm 4. We have that $v(\mathbf{e}) \leq S$, with the equality holding when the worker is a qualified worker. Each task submitted by this worker is worth $v(\mathbf{e})$ to the platform, minus the cost of labor $r(\mathbf{e})$. When the offered wage for a qualified worker is r, the wage for a worker with confusion matrix $\mathbf{e}$ is $r(\mathbf{e}) = r \cdot \frac{v(\mathbf{e})}{S}$. Given that the worker will not work when the offered wage $r(\mathbf{e})$ is lower than the reservation wage w, the net profit from this worker is:

$$\mathit{Profit}(r, \mathbf{e}, w, t) = \begin{cases} 0 & : w > r\frac{v(\mathbf{e})}{S} \\ (v(\mathbf{e}) - r\frac{v(\mathbf{e})}{S}) \cdot t & : w \leq r\frac{v(\mathbf{e})}{S} \end{cases} \tag{21}$$

where t is the lifetime of the worker. Integrating this over all possible values of error rates $\mathbf{e}$, reservation wages w, and worker lifetimes t, the expected net profit of the platform is:

$$\begin{aligned} \mathit{Profit}(r) &= \int_0^\infty \int_0^\infty \int_{\mathbf{e}} \mathit{Profit}(r, \mathbf{e}, w, t) \cdot f(\mathbf{e}, w, t) \mathrm{d}\mathbf{e} \mathrm{d}t \mathrm{d}w \\ &= \int_0^r \int_0^\infty \int_{\mathbf{e}} (v(\mathbf{e}) - r\frac{v(\mathbf{e})}{S}) \cdot t \cdot f(\mathbf{e}, w, t) \mathrm{d}\mathbf{e} \mathrm{d}t \mathrm{d}w \end{aligned} \tag{22}$$

The optimal price r^* is given by the solution to the maximization problem:

$$r^* = \arg\max_r \mathit{Profit}(r) = \arg\max_r \int_0^r \int_0^\infty \int_{\mathbf{e}} (v(\mathbf{e}) - r \cdot \frac{v(\mathbf{e})}{S}) \cdot t \cdot f(\mathbf{e}, w, t) \mathrm{d}\mathbf{e} \mathrm{d}t \mathrm{d}w \tag{23}$$

When distribution $f(\mathbf{e}, w, t)$ is known, r^* can be computed through a variety of optimization methods, returning r^*, the optimal price to pay for a qualified worker. Then, the optimal price for an unqualified worker is given by $r^* \frac{v(\mathbf{e})}{S}$.

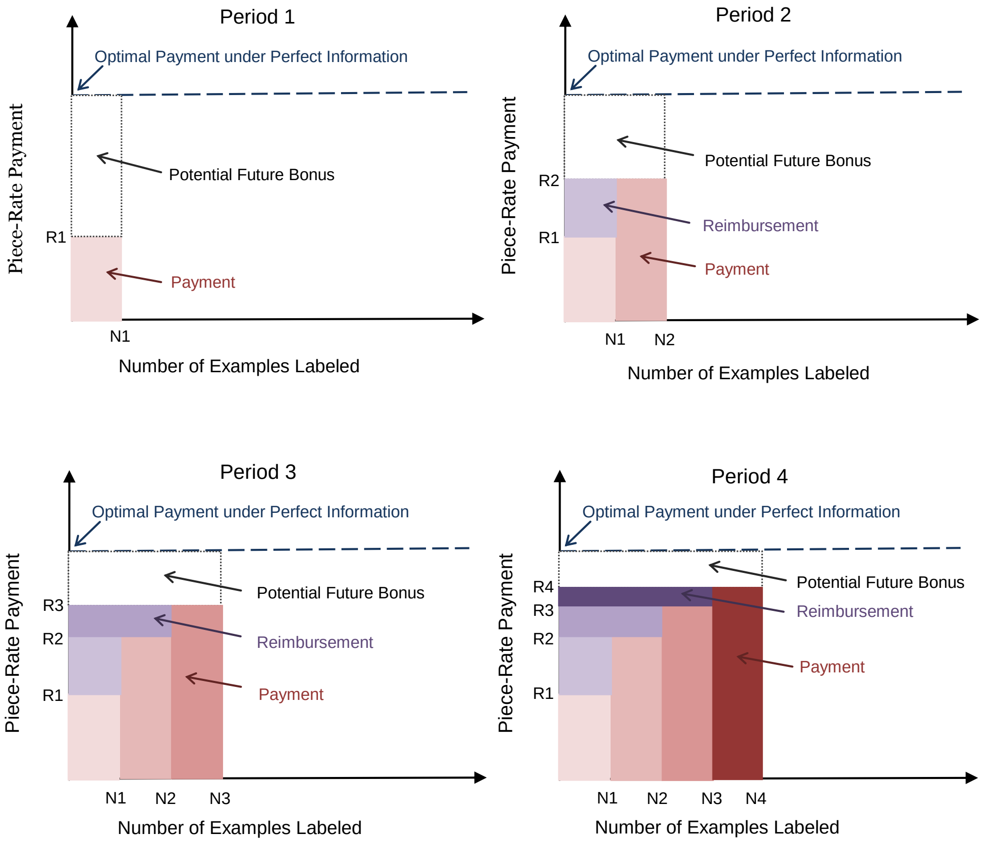


Figure 4: An illustration of Real-Time Payment

7.3 Real-time Pricing under Imperfect Knowledge of Worker Quality

The estimates we get following the techniques in Section 5.1 are deterministic, but imperfect. Although our model assumes that the true quality of a worker **e** is fixed, our *estimate* of **e** is changing over time. Just based on sampling theory, the piece-rate payment of a worker is expected to fluctuate as the worker labels more examples, even if the payment is expected to converge towards the optimal price over time. Unfortunately, this fluctuation is not an acceptable part of a payment scheme. A worker would be negatively surprised if suddenly their perfect wage plummets just due to a single labeling mistake. How to pay workers in this setting?

Ideally, we want a payment scheme that:

1. Rewards workers with a payment that is as close as possible to their (unknown) optimal price.
2. Avoids payment fluctuations, resulting from expected measurement fluctuations, preferring a

smooth payment evolution over time.

3. Avoids a decreasing payment slope, which can be interpreted as punishment, and prefer payment schemes that have either stable or increasing payment slopes.

Condition 1 allows for maximum worker engagement: Each worker has a reservation wage and a lifetime: if the average piece-rate payment at the end of lifetime is lower than the reservation wage, the worker will not participate in the task. Of course, the more examples a worker labels, the closer the payment $r\hat{(\mathbf{e})}$ is to the optimal payment $r(\mathbf{e})$ under perfect knowledge of worker quality. Unfortunately, this scheme also leads to significant up and down fluctuations (violating condition 2), especially early on, leading to worker confusion. To avoid the sudden fluctuations, we can pay based on a moving average of worker quality, which softens the potential estimation fluctuations. Unfortunately, paying using a moving average can also lead to a decrease in payment over time, if the worker starts by giving a few correct answers before naturally reverting back to the mean performance.

Our solution is a process that we call "*payment with reimbursements*". Our scheme rewards workers over time by paying based on pessimistic estimates of worker quality (i.e., underpays initially) but compensates for the underpayment by *reimbursing* in later periods the payment withheld due to the uncertainty. To ensure a pessimistic estimate of quality, we impose low prior on the Bayesian estimation of the worker quality, assuming that the worker has an average quality that generates a very low payment. When a non-spammer worker submits answers, the distribution of quality increases, allowing the payment to increase over time. Then, as we get more data, we proceed with the payment estimations, reimbursing the workers for the underpayment in the prior periods. Given that payment over time is effectively a sum of random variables, Chernoff's bound applies in this case, guaranteeing that the uncertainty of payment decreases exponentially with the number of tasks submitted; therefore our payment scheme converges into the real payment with exponentially low probability of overpaying.

Figure 4 illustrates the process: for a worker, we divide his lifetime into a set of small periods (for example, paying every 10 completed tasks). At the end of each period, we first pay the worker the deserved earnings in the current period, then reimburse the worker for the price difference between this period and the previous period for all the tasks completed before this period, as illustrated by Figure 4. Suppose that the piece-rate payment after the first 10 submissions is R_1, the worker gets paid $10 \cdot R_1$ at the end of Period 1. Now, the piece-rate payment after the second 10 submissions is R_2, we first pay the worker $10 \cdot R_2$ and then reimburse the "unpaid" part for the first 10 submissions by the price difference $10 \cdot (R_2 - R_1)$. Similarly, in the third period we examine if there are "unreimbursed" payments for the first and second periods, and do the same. We repeat the process until the worker reaches the end of the lifetime.

Notice that the strategy has the fortunate side-effect of incentivizing long-term participation: At any given time point, the worker improves payment by: (a) increasing the estimated pay rate $\hat{r}(\mathbf{e})$ (and bringing it closer to optimal payment $r(\mathbf{e})$), and (b) receiving a reimbursement payment (phrased as "bonus" to the worker) for all the underpayments in the prior periods. This strategy encourages good workers to work more, allowing us to understand better their quality. On the contrary, a worker that does not plan to work for long (therefore imposing to the platform the risk of handling the unknown quality of the worker), receives a comparatively lower payment for the same amount and quality of work. So each incoming worker goes through a "reputation building" stage during which she is likely to be underpaid. However, as she completes more and more tasks, we will know better about her true quality and her payment will then increase.

8 Simulation Results

In this section, we conducted a set of simulations to test the performance of the strategies proposed earlier in the paper. In Sections 8.1 and 8.2 we present results that indicate the effectiveness of estimating worker quality, and the ability of the proposed techniques to achieve the desired data quality in the most cost-efficient manner. Then, in Section 8.3, we present an analysis of the performance of our pricing strategy, illustrating how it outperforms existing, simpler baselines.

8.1 Effectiveness in Estimating Worker Quality

The success of the EM algorithm pivots on the accurate estimation of worker quality. In traditional testing environments, "gold" labels for the objects are always available, whereas for EM, the true labels for the objects are unknown. A natural question to raise is: how well does EM perform in estimating the true quality of workers, compared with gold testing?

We did some simulations to test the effectiveness of the EM algorithm. There is a set of objects, with true labels randomly and equally generated from two categories. We have 1000 simulated

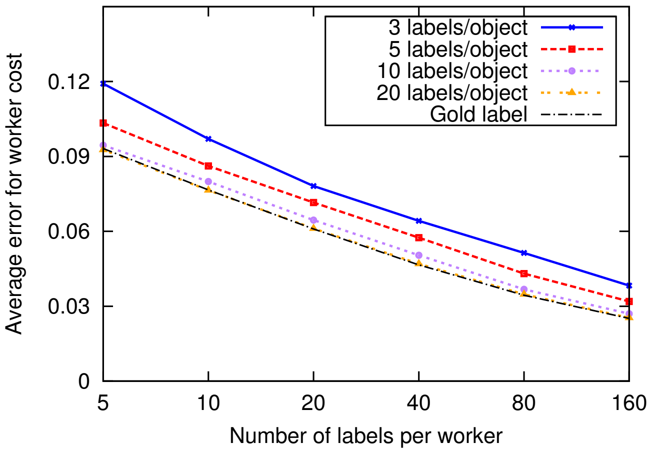


Figure 5: The performance of the EM algorithm in estimating worker quality

workers, each of whose confusion matrix $\mathbf{e}$ is drawn from a set of two beta distributions: $\mathbf{Beta(4, 2)}$ and $\mathbf{Beta(2, 4)}$, each corresponding to a row of the confusion matrix $\mathbf{e}$. Each label assigned by the worker is generated according to the true class of the object and the confusion matrix of the worker. We employed the EM algorithm to estimate the true labels of the objects as well as the quality of workers at the same time. Since the quality of the data depends on the redundancy level (i.e., the number of workers assigned to provide labels), we used four treatments: 3 workers per object, 5 workers per object, 10 workers per object, 20 workers per object. The accuracy of worker quality estimation is measured by the absolute deviation of the estimated cost of the worker from her true cost. The lower the deviation, the better the estimation. We consider the simplest case of misclassification cost: a cost of 1 is incurred when an object is misclassified, and 0 otherwise.

Figure 5 shows the results for average error in estimating worker cost[22]. The average cost estimation error for gold testing serves as a lower bound, indicated by the black dashed line. As expected, the accuracy of quality estimation improves as workers label more objects: The more you test, the more confident you are. Redundancy level also plays an important role: The more redundancy there is, the closer EM estimation is to gold testing. Actually, when there are 20 workers assigned for each object, the performance of the EM algorithm is as good as gold testing. Notice that, in our simulation, the overall worker quality is pretty low. The same equivalency can be achieved with lower redundancy if the quality of workers is relatively high.

[22] The x-axis is shown in log-scale.

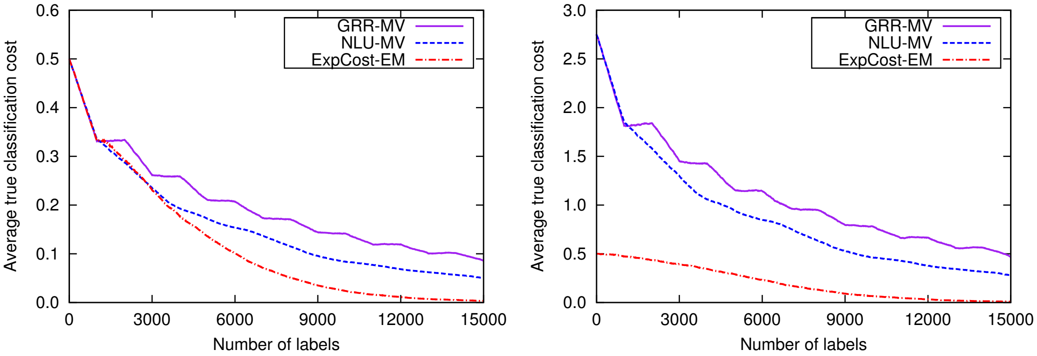
8.2 Effectiveness in Achieving Data Quality

In the previous section, we examined the ability of the expectation maximization algorithm to estimate worker quality. Of course, knowing the quality of the workers is just a means towards achieving a high-quality labeling of the data. The goal is to label data points with the target SLA-promised quality, using as few workers as possible. Our approach prioritizes data points for labeling based on their *expected misclassification cost* (see Section 6.1). We refer to our approach as *ExpCost*. We compare our *ExpCost* method with the current state-of-the-art approaches, namely a round-robin strategy (*GRR*) that assigns the same level of effort in objects, and a selective labeling strategy (*NLU*) from Ipeirotis et al. (2013), that puts priority on objects with high uncertainty (without considering the quality of individual workers explicitly). In both *GRR* and *NLU*, the final class is determined using simple majority voting (MV) since the methods are agnostic to differences in worker quality when labeling the same item. In contrast, in *ExpCost*, the final class is determined using weighted majority voting, as discussed above in the context of the EM algorithm.

The simulation setup is as follows: we have 1000 objects, evenly assigned to two categories, and 200 workers. We draw the confusion matrix $\mathbf{e}$ of each worker from a set of two beta distributions: $\mathbf{Beta(4, 2)}$ and $\mathbf{Beta(2, 4)}$, each corresponding to a row of the confusion matrix $\mathbf{e}$. Each time, we draw a worker uniformly from the worker pool, and depending on the strategy used (*GRR*, *NLU*, and *ExpCost*), we assign the worker to the example with the highest priority. We test the performance of our proposed method under two settings: a symmetric cost matrix $\mathbf{c}^{(a)}$, and an asymmetric cost matrix $\mathbf{c}^{(b)}$.

$$\mathbf{c}^{(a)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad ; \quad \mathbf{c}^{(b)} = \begin{pmatrix} 0 & 1 \\ 10 & 0 \end{pmatrix}$$

Figure 6 shows the actual misclassification cost for the data as a function of the number of labels acquired for *GRR*, *NLU*, and *ExpCost*, under the two cost settings. The first observation is that the *ExpCost* method beats both *GRR* and *NLU* consistently. The advantage becomes even more substantial when classification cost are asymmetric. Second, in Figure 6(a), *NLU* and *ExpCost* have a similar performance during the early stages (when the number of labels acquired is less than 4000): this happens because the EM algorithm has not obtained good estimates for workers yet. The performance gap between *ExpCost* and *NLU* increases later on, showing that knowing the individual



(a) symmetric cost matrix $c_{ij}^{(a)}$

(b) asymmetric cost matrix $c_{ij}^{(b)}$

Figure 6: Average true classification cost as a function of the number of labels acquired, for a round robin strategy (*GRR*), a selective labeling strategy based on label uncertainty (*NLU*), and our proposed strategy based on expected cost (*ExpCost*)

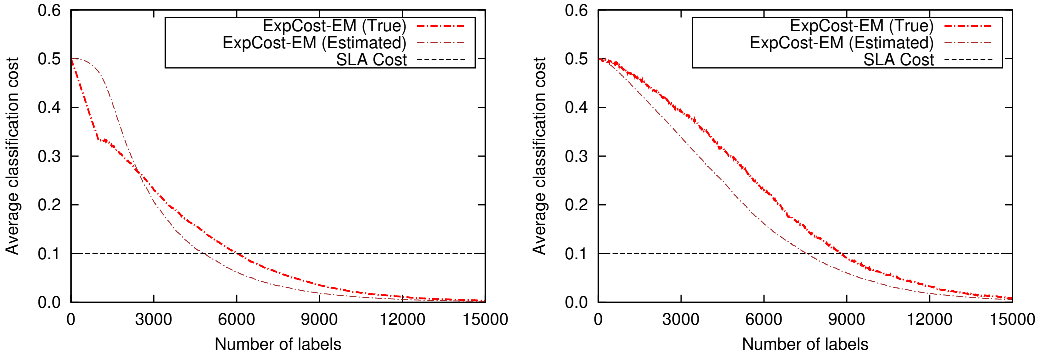
worker quality can help obtaining better data quality.[23]

Another way to interpret the results in Figure 6 is that the *ExpCost* method can achieve the SLA quality in a much more resource-efficient manner. Assume that the SLA sets $\tau_c \leq 0.1$. For case (a), we need to acquire 14090 labels if *GRR* is used, 8730 labels if *NLU* is used, and only 6030 labels if *ExpCost* is used; for case (b), we need to acquire more than 15000 labels for both *GRR* and *NLU*, and only 8770 labels if *ExpCost* is used. This reduction in the required number of labels has the potential to lower the cost we need to pay for acquiring labels from crowdsourcing workers.

The evaluations above is based on the actual (true) cost of misclassification, which requires the knowledge of true labels of the objects. One practical problem arises from the difficulty of knowing when the SLA quality is achieved. How close is the estimated cost of misclassification to the true cost? Figure 7 shows both the true cost and estimated cost of misclassification for *ExpCost* strategy, together with the target cost specified in SLA. In general, the estimated cost tends to misestimate the classification cost, especially in the early period of estimation when the overall data quality is not very good. But as the number of labels increase, the estimated cost is getting close to the true cost.[24].

[23] For all the later experiments, we use *ExpCost* method for label resource allocation.

[24] In practice, another approach is to use acceptance sampling to determine whether the quality of the data reaches SLA.



(a) symmetric cost matrix $c_{ij}^{(a)}$ (b) asymmetric cost matrix $c_{ij}^{(b)}$

Figure 7: Average true vs. estimated classification cost as a function of the number of labels acquired for *ExpCost*. The red and brown lines represent the true and estimated cost repectively, and the dotted black line defines the SLA target cost

8.3 Optimal Pricing Strategy

We also conducted a set of synthetic experiments to test the performance of our proposed pricing strategy. We describe below the setting for the synthetic experiments.

We assume that the platform receives $N = 10,000$ labeling tasks from a client, who is willing to pay $S = 200$ for each successfully completed task. We assume that the task is a binary classification problem. The SLA requirement sets the expected misclassification cost at $\tau_c \leq 0.01$.

- The platform is interested in optimizing profit per time unit, which is given by

$$\frac{N \cdot S - \sum_{(k)} r(\mathbf{e}^{(k)}) \cdot t^{(k)})}{T}$$

  where $r(\mathbf{e}^{(k)})$ refers to the reward given to worker k under the pricing strategy, and T is the total time needed to complete the task.

- The requester announces a price scheme and distributes tasks in batches. The size of each batch is $N_{Batch} = 100$ tasks.
  - When a worker arrives, she is assigned to the example with the highest expected cost in the current batch.
  - If the average expected cost of the current batch is lower than the SLA requirement, the platform releases the data to the client, and moves to the next batch. We denote the time

of this event by T.

- Every 600 time units, 10 new workers come into the marketplace. For each worker k, the confusion matrix $\mathbf{e}^{(k)}$, reservation wage $w^{(k)}$, and lifetime $t^{(k)}$ are drawn from a distribution known to the platform.
  - The worker sees the announced (possibly quality-based) price and computes her expected piece-rate payment after completing $t^{(k)}$ tasks.
  - If the expected payment is lower than her reservation wage $w^{(k)}$, the worker leaves the market. Otherwise, the worker submits tasks at a speed of one task per $s = 30$ time units.
  - The worker stops working either because the task finished, or because the worker reached the maximum lifetime point.

The confusion matrix, reservation wage, and lifetime of a worker are generated following the procedure below:

- Draw v_e, v_w and v_t from a trivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
- Transform v_e, v_w, v_t to $\mathbf{e}$, w, t by setting: $e_{00} = e_{11} = 0.5 + 0.5 \cdot \text{logit}^{-1}(v_e)$, $w = \exp(v_w)$, $t = \exp(v_t)$.

The parameters for the trivariate normal distribution are given below. (For now, we assume that the lifetimes of the workers are independent of quality and reservation wage).

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} 0.0 \\ 2.0 \\ 5.0 \end{pmatrix} \quad ; \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{pmatrix} = \begin{pmatrix} 1.0 & \rho_{12} & 0.0 \\ \rho_{12} & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

We test our strategy under two values of ρ_{12}, changing the correlation between worker reservation wage and worker quality:

- $\rho_{12} = 0.0$: the quality of the workers and their reservation wages have no correlation.
- $\rho_{12} = 0.8$: the quality of the workers and their reservation wages are positively correlated.

Figures 8(a) and (b) show the scatter plot and the two histograms along x-axis and y-axis for $\rho_{12} = 0.0$ and $\rho_{12} = 0.8$, respectively.[25] As we can see from the plots, when $\rho_{12} = 0.0$ the reservation wage is independent of the level of quality, while when $\rho_{12} = 0.8$ the reservation wage tends to be higher as the quality level of workers increases.

Results: We compared our pricing strategy with *uniform pricing*, where all workers receive the same wage. We experimented with a wide range of price values from 4.4 to 12.5 (i.e, 30%, 40%, 50%, 60%, and 70%- quantile of the reservation wage distribution, respectively). Figure 9 shows the average profits across different time points for $\rho_{12} = 0.0$ and $\rho_{12} = 0.8$. The two solid lines represent the quality- based pricing. At the end points, our strategy outperforms the best uniform pricing strategy by 24.6% when there is no correlation between worker quality and reservation age, and 159.6% when worker quality and reservation wage are positively correlated.

9 Managerial Implications and Limitations

9.1 Managerial Implications

In our work, we presented a holistic approach for managing and paying crowdsourced workers, that reduces the need for testing and seamlessly combines testing with production. The methods described in this paper have been implemented and are available as open source at [removed for anonymity]. Our toolkit has been deployed in practice in multiple industrial applications, and has been used to manage tens of thousands of crowdsourced workers over the last couple of years. As mentioned in Section 3, our work serves as a fundamental quality control block, for a variety of tasks, ensuring that the outcome of crowdsourced production reaches the quality levels required by the employers.

Crowdsourcing is rapidly becoming a commonly used tool across many Fortune-500 companies. Amazon has been using paid crowdsourcing for more than 10 years now to de-duplicate products in the catalogs uploaded to their platform by merchants. Metaweb (acquired by Google in 2010) has been using paid crowdsourcing in order to create Freebase (currently the Google Knowledge Graph) (Kochhar et al., 2010). Microsoft has built the Microsoft Universal Human Relevance System (UHRS)[26] in order to evaluate and improve the results in Bing, their search engine. Facebook is

[25] Both are generated using a sample size of 5000 randomly drawn workers.

[26] http://www.signalprocessingsociety.org/technical-committees/list/sl-tc/spl-nl/2013-05/interview-crowdsourcing/

using crowdsourcing for content moderation, and Twitter is using Mechanical Turk[27] to improve their real time event detection, and many other companies (e.g., see http://www.crowdconf.com/) use crowdsourcing either directly or through an intermediary. Firms are attracted to crowdsourcing due to the dynamic nature of hiring, allowing quick scaling up and quick downsizing of the workforce, according to the needs of the company, with reaction times within hours or even minutes.

Unfortunately, quality control remains an issue and most existing solutions simply attempt to simply screen workers through multiple gold tests, and reject unqualified workers. Allowing for more fine-grained strategies can allow for lowering the barriers-to-entry for firms, and allowing the higher use of crowdsourcing. The wider adoption of crowdsourcing can also lower the barrier-to-enter for workers with no prior experience and reputation. Since there is no interview stage, and workers can join the workforce at-will, it becomes easier for unemployed people to find work and prove their skills while working. Since our approach can automatically provide a performance measurement for each worker, we can also lower the barrier for providing honest reputation feedback, that can facilitate the creation of a healthy, well-operating crowdsourcing marketplace. Our pricing scheme further ensures that workers are paid according to the quality they contribute, incentivizing employers to open more of their tasks to "crowd workers." Since our pricing schemes ensures an (eventually-)fair payment policy, good workers are also encouraged to keep working for long periods of time, which reduces the churning of good workers—one of the big problems for any employer, and particularly acute one in crowdsourcing.

9.2 Limitations

Of course, there are limitations to this work and corresponding opportunities for further research:

- We assume that the qualities of workers are independent. In practice, workers might have correlated errors, either positively or negatively, which would certainly affect the validity of our algorithm. However, not all correlations are harmful. Previous research (Kuncheva et al., 2003; Clemen and Winkler, 1985) has show that negative correlation between workers could increase the accuracy of classification results, but positive correlation can result in lower labeling quality than expected.

- The quality of workers might change over time. For many types of tasks, there might be either

[27] http://engineering.twitter.com/2013/01/improving-twitter-search-with-real-time.html

learning effects or deterioration effects, which makes the quality of workers vary across time. To account for the potential instable nature of worker quality, we can apply particle filtering method to track the change in worker quality over time (Crisan and Doucet, 2002; Donmez et al., 2010).

- We use the concept of reservation wage, which may or may not be a model directly translatable into labor marketplaces. For example, Horton and Chilton (2010) showed that workers on Mechanical Turk tend to exhibit behaviors of target earning, and stop working when they meet their daily income target.
- We assume that the platform owner knows or can estimate relatively well the joint distribution of worker qualities, reservation wage, and lifetimes. In reality, the platform needs to learn this distribution, especially in an environment where workers arrive and leave the platform. This estimation task should be studied carefully across real labor marketplaces.
- We do not report experiments with real workers in this paper. This is important future work, but producing a high-quality experimental study of how workers react to different quality control schemes and incentives is beyond what we could reasonably do in a single paper.

Despite these limitations, we believe that our current work provides a solid foundation on which future work can build. Furthermore, our work can be used immediately by interested parties, allowing for easier management of crowdsourced workers, and therefore the development of further interesting applications, enabled by crowdsourcing.

References

Agranov, Marina, Chloe Tergiman. 2012. Incentives and compensation schemes: An experimental study. *International Journal of Industrial Organization* .

Akerlof, G. A. 1970. The market for ‘lemons’: Quality uncertainty and the market mechanism. *Quarterly Journal of Economics* **84**(3) 488–500.

Bachrach, Yoram, Thore Graepel, Tom Minka, John Guiver. 2012. How to grade a test without knowing the answers—a bayesian graphical model for adaptive crowdsourcing and aptitude testing. *arXiv preprint arXiv:1206.6386* .

Berger, Roger L. 1982. Multiparameter hypothesis testing and acceptance sampling. *Technometrics* **24**(4) 295–300.

Bernstein, M. S., G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, K. Panovich. 2010. Soylent: A word processor with a crowd inside. *Proceedings of the 23th annual ACM Symposium on User Interface Software and Technology*. 313–322.

Carpenter, B. 2008. Multilevel Bayesian models of categorical data annotation. Available at http://lingpipe-blog.com/lingpipe-white-papers/.

Clemen, Robert T, Robert L Winkler. 1985. Limits for the precision and value of information from dependent sources. *Operations Research* **33**(2) 427–442.

Crisan, Dan, Arnaud Doucet. 2002. A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on* **50**(3) 736–746.

Crocker, Linda, James Algina. 2006. *Introduction to Classical and Modern Test Theory*. Wadsworth.

Dawid, A. P., A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics* **28**(1) 20–28.

DeMars, Christine. 2010. *Item Response Theory*. Oxford University Press.

Dodge, Harold F, American Society for Quality Control. 1973. *Notes on the Evolution of Acceptance Sampling*. American Society for Quality Control.

Donmez, Pinar, Jaime Carbonell, Jeff Schneider. 2010. A probabilistic framework to learn from multiple annotators with time-varying accuracy. *SIAM International Conference on Data Mining (SDM)*. 826–837.

Gale, Douglas, Martin Hellwig. 1985. Incentive-compatible debt contracts: The one-period problem. *The Review of Economic Studies* **52**(4) 647–663.

Horton, John Joseph, Lydia B Chilton. 2010. The labor economics of paid crowdsourcing. *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 209–218.

Ipeirotis, P. G. 2010. Analyzing the Amazon Mechanical Turk marketplace. *XRDS* **17** 16–21.

Ipeirotis, Panagiotis G., Foster Provost, Victor Sheng, Jing Wang. 2013. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery* Forthcoming. Available at SSRN: http://ssrn.com/abstract=1688193.

Ipeirotis, Panagiotis G, Foster Provost, Jing Wang. 2010. Quality management on amazon mechanical turk. *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 64–67.

Kittur, A., B. Smus, S. Khamkar, R. E. Kraut. 2011. CrowdForge: Crowdsourcing complex work. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. 43–52.

Kochhar, Shailesh, Stefano Mazzocchi, Praveen Paritosh. 2010. The anatomy of a large-scale human computation engine. *Proceedings of the ACM SIGKDD Workshop on Human Computation*. ACM, 10–17.

Kulkarni, Anand P., Matthew Can, Bjoern Hartmann. 2011. Turkomatic: Automatic recursive task and workflow design for mechanical turk. *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*. 2053–2058.

Kuncheva, L. I., C. J. Whitaker, C. A. Shipp, R. P. W. Duin. 2003. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications* **6**(1) 22–31.

Lazear, Edward P. 1986. Salaries and piece rates. *Journal of Business* 405–431.

Lazear, Edward P. 2000. Performance pay and productivity. *American Economic Review* 1346–1361.

Lazear, Edward P, Sherwin Rosen. 1979. Rank-order tournaments as optimum labor contracts.

Little, G., L. B. Chilton, M. Goldman, R. Miller. 2010. Turkit: Human computation algorithms on mechanical turk. *Proceedings of the 23th annual ACM Symposium on User Interface Software and Technology*. 57–66.

Malone, T. W., R. Laubacher, C. Dellarocas. 2010. Harnessing crowds: Mapping the genome of collective intelligence. Available at http://ssrn.com/abstract=1381502.

Meyer, Margaret A, John Vickers. 1997. Performance comparisons and dynamic incentives. *Journal of Political Economy* **105**(3) 547–581.

Mookherjee, Dilip. 1984. Optimal incentive schemes with many agents. *The Review of Economic Studies* **51**(3) 433–446.

Nov, Oded, Ofer Arazy, David Anderson. 2011. Dusting for science: motivation and participation of digital citizen science volunteers. *Proceedings of the 2011 iConference*. iConference '11, 68–74. doi: 10.1145/1940761.1940771. URL http://doi.acm.org/10.1145/1940761.1940771.

Raykar, Vikas C, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, Linda Moy. 2010. Learning from crowds. *The Journal of Machine Learning Research* **99** 1297–1322.

Resnick, Paul, Ko Kuwabara, Richard Zeckhauser, Eric Friedman. 2000. Reputation systems. *Communications of the ACM* **43**(12) 45–48.

Schilling, Edward G. 1982. *Acceptance Sampling in Quality Control*, vol. 42. CRC PressI Llc.

Sheng, Victor S, Foster Provost, Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 614–622.

Snow, Rion, Brendan O'Connor, Daniel Jurafsky, Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 254–263.

Spulber, Daniel F. 1996. Market microstructure and intermediation. *The Journal of Economic Perspectives* **10**(3) 135–152.

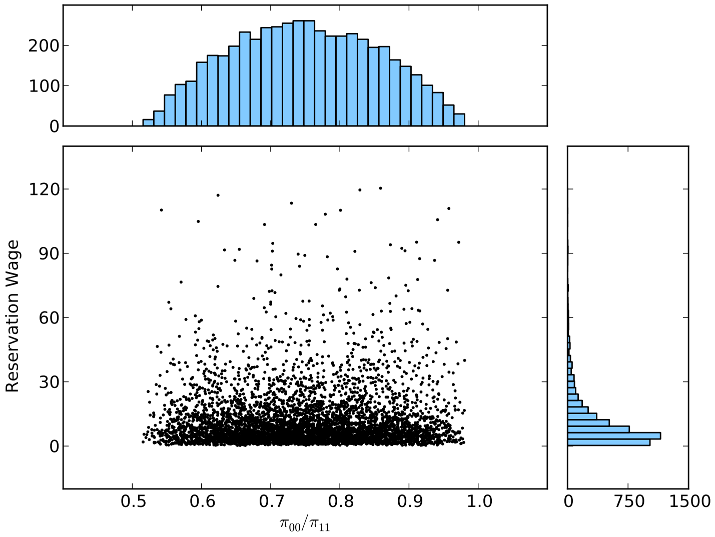
Stewart, Osamuyimen, David Lubensky, Juan M. Huerta. 2010. Crowdsourcing participation inequality: a scout model for the enterprise domain. *Proceedings of the ACM SIGKDD Workshop on Human Computation*. HCOMP '10, 30–33. doi:10.1145/1837885.1837895. URL http://doi.acm.org/10.1145/1837885.1837895.

Townsend, Robert. 1979. Optimal contracts and competitive markets with costly state verification. *Journal of Economic theory* **21**(2) 265–293.

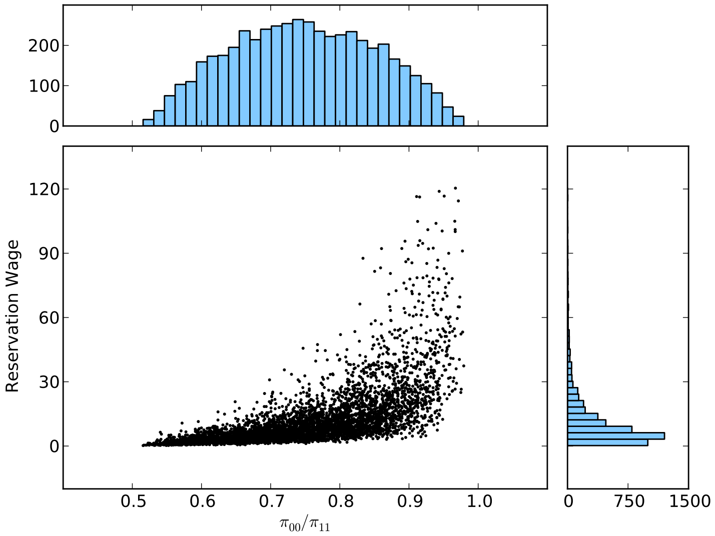
Welinder, Peter, Steve Branson, Serge Belongie, Pietro Perona. 2010. The multidimensional wisdom of crowds. *Advances in Neural Information Processing Systems* **23** 2424–2432.

Wetherill, GB, WK Chiu. 1975. A review of acceptance sampling schemes with emphasis on the economic aspect. *International Statistical Review/Revue Internationale de Statistique* 191–210.

Whitehill, J., P. Ruvolo, T. Wu, J. Bergsma, J. Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in Neural Information Processing Systems 22 (NIPS 2009)*. 2035–2043.

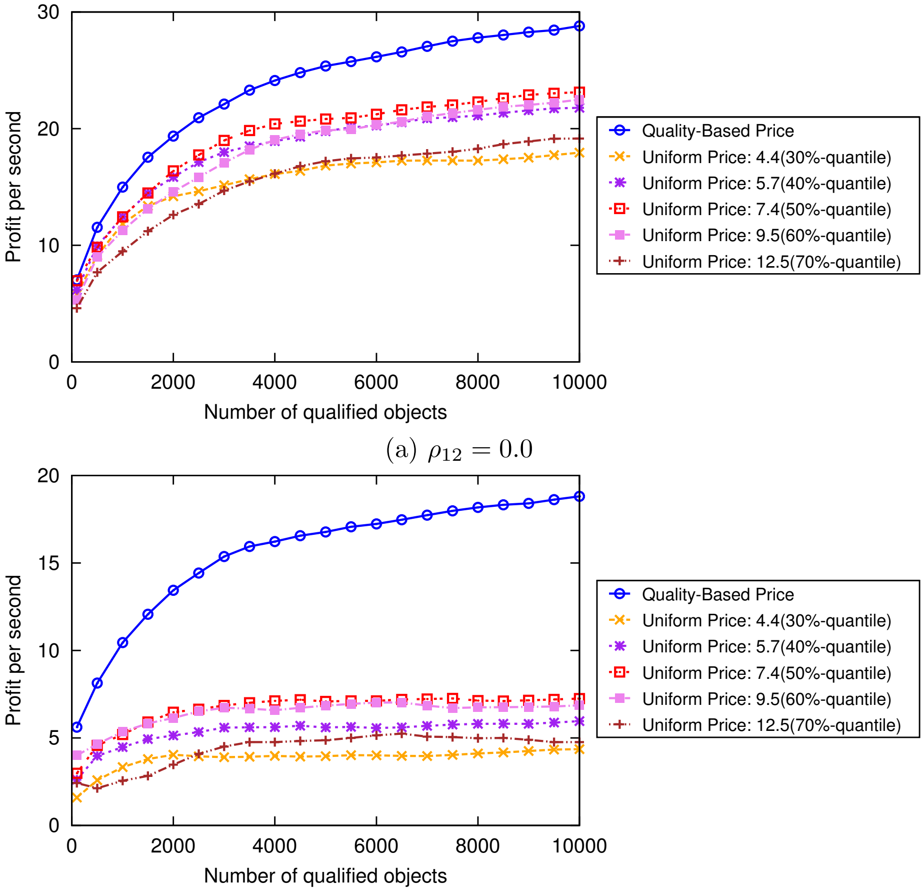


(a) $\rho_{12} = 0.0$



(b) $\rho_{12} = 0.8$

Figure 8: Histograms of Parameter Values



(a) $\rho_{12} = 0.0$

(b) $\rho_{12} = 0.8$

Figure 9: Average profits per second over time for different pricing schemes