Cost-Effective Quality Assurance in Crowd Labeling

Jing Wang*[1], Panagiotis G. Ipeirotis†[2], and Foster Provost‡[2]

[1]School of Business and Management, Hong Kong University of Science and Technology

[2]Leonard Stern School of Business, New York University

Abstract

The emergence of online *paid micro-crowdsourcing* platforms, such as Amazon Mechanical Turk (AMT), allows on-demand and at scale distribution of tasks to human workers around the world. In such settings, online workers come and complete small tasks posted by employers, working for as long or as little as they wish, a process that eliminates the overhead of the hiring (and dismissal). 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. A number of research challenges arise in such settings. How can we ensure that the submitted work is accurate? What allocation strategies can be employed to make the best use of the available labor force? How to appropriately assess the performance of individual workers? In this paper, we consider labeling tasks and develop a comprehensive scheme for managing the quality of crowd labeling: First, we present several algorithms for inferring the true classes of objects and the quality of participating workers, assuming the labels are collected all at once before the inference. Next, we allow employers to adaptively decide which object to assign to the next arriving worker and propose several heuristic-based dynamic label allocation strategies to achieve the desired data quality with significantly fewer labels. Experimental results on both simulated and real data confirm the superior performance of the proposed allocation strategies over other existing policies. Finally, we introduce two novel metrics that can be used to objectively rank the performance of crowdsourced workers, after fixing correctable worker errors and taking into account the costs of different classification errors. In

*jwang@ust.hk

†panos@stern.nyu.edu

‡fprovost@stern.nyu.edu

particular, the worker value metric directly measures the monetary value contributed by each label of a worker towards meeting the quality requirements and provides a basis for the design of fair and efficient compensation schemes.

Keywords: crowd labeling, quality assurance, dynamic label allocation, worker performance metric

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 posted on platforms like Amazon Mechanical Turk[1] (AMT) to big innovation contests conducted by Netflix[2] and Innocentive[3]. AMT, in particular, dominates today the market for crowdsourcing micro-tasks, which are easy for humans to accomplish, but remain challenging for computers (Ipeirotis, 2010). The employers on AMT can post a variety of small tasks, such as image tagging, sentiment judgment, language translation, and text annotation. Workers complete these tasks and get compensated in the form of micro-payments, typically in the range of 5 to 20 cents per task. The immediate and elastic supply of cheap labor in micro-crowdsourcing systems makes it possible to complete tasks at low cost and with high throughput.

Firms, ranging from Fortune 500 companies to small startups, are increasingly attracted to micro-crowdsourcing to meet their business needs. Amazon has been using micro-crowdsourcing for more than 10 years to de-duplicate products in catalogs uploaded to its platform by merchants. Microsoft has built the Universal Human Relevance System (UHRS)[4] to evaluate and improve the performance of its search engine—Bing. Facebook has been relying on micro-crowdsourcing for content moderation, and Twitter is using AMT to improve its real-time event detection accuracy.[5] Many other companies employ micro-crowdsourcing either directly or through an intermediary (e.g., AMT, CrowdFlower, CrowdSource).

Micro-crowdsourcing platforms are also increasingly used by IS researchers for a wide variety of data labeling and annotation tasks. To study the impact of review text on product sales, Archak et al. (2011) recruited workers from AMT to extract product features and opinions about these features from the text of product reviews. Moreno and Terwiesch (2014) used AMT workers to code the sentiment of comments left by previous service buyers as either positive or negative and constructed a reputation measure for service providers based on this information. Wang et al. (2012) also relied on AMT platform to obtain a reliable measure of the perceived helpfulness of user-generated reviews.

[1] https://www.mturk.com/
[2] http://www.netflixprize.com/
[3] http://www.innocentive.com/
[4] http://www.signalprocessingsociety.org/technical-committees/list/sl-tc/spl-nl/2013-05/interview-crowdsourcing/ (Accessed November 14th, 2015.)
[5] http://blog.echen.me/2013/01/08/improving-twitter-search-with-real-time-human-computation/(Accessed November 14th, 2015.)

Despite the promise, significant challenges remain. Workers in micro-task crowdsourcing markets usually have different levels of expertise and dedication and thus exhibit heterogeneous quality in task execution. Unfortunately, verifying the quality of every submitted answer is an expensive operation and negates many advantages of micro-crowdsourcing: the cost and time for verifying the correctness of each submitted answer (e.g., checking the answers for a question such as "Do you see any recognizable human face in the picture?") are typically comparable to the cost and time for performing the task itself. The difficulty of verification makes micro-crowdsourcing systems prone to errors, which harms the reliability, scalability, and robustness of such markets (Wais et al., 2010).

Our main research objective is to develop a comprehensive scheme for assuring the quality of micro-task crowdsourcing in a cost-effective manner. In this paper, we focus on the quality control of binary labeling tasks (e.g., "Does this photograph violate the terms of service? Yes or No."). While this might seem limiting, we show in Appendix A that many complex tasks can be broken down into a set of simpler operations for which a binary choice task serves as a key building block for quality assurance. Hence, our proposed scheme naturally fits into such workflows and provides a fundamental quality control mechanism for other more complicated operations. Such synergies lead to workflows that can accomplish complex tasks with guarantees of high-quality output, even when the underlying workforce has uncertain, varying, or even moderate-to-low quality.

In the crowd labeling settings, one common approach used by employers to ensure reliability is to introduce redundancy: ask multiple workers to work on the same task and infer the correct answer using some aggregation method such as majority voting. In this paper, we are interested in the following optimization problem: *Suppose an employer wishes to achieve a certain level of data labeling quality, what strategies can she use to minimize the expense of acquiring labels from crowdsourced workers?*

The decision system in our framework consists of two phases: a *label allocation* phase and an *inference* phase. In the label allocation phase, the unlabeled or partially labeled objects are assigned to crowdsourced workers for labeling. In the inference phase, an algorithm is used to infer the true classes of objects. The system is called *static* if the inference phase begins after the full completion of the allocation phase. In a static system, labels are allocated all at once before the inference process starts. Most of the previous studies assume one-time allocation of labels and devote the research effort to improving the inference accuracy (e.g., Whitehill et al., 2009; Raykar et al., 2010; Welinder et al., 2010; Karger et al., 2011). The system is called *dynamic* if the two phases are interleaved. A dynamic system iterates over these two phases until the desired data quality is achieved or the available resources are exhausted, allowing the employer to adaptively decide which object to assign to the next arriving worker based on the stream of collected labels so far. Figure 1 illustrates these two decision systems.

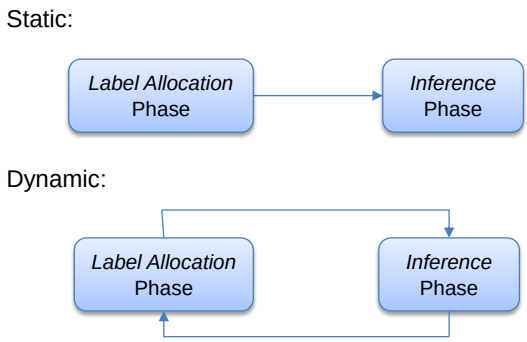


Figure 1: The static and dynamic decision systems

The characteristics of micro-task crowdsourcing platforms make them well suited for the implementation of a dynamic decision system. Workers arrive in the market over time, and once they agree to work on the tasks, they label the objects one after another. The decision about which object to assign to the worker next can be postponed until she finishes labeling the current object. Since the allocation decision can be made within a very short period of time, the worker will not feel any latency in waiting for the next labeling task. As will be shown later, the possibility to make label assignment on the spot instead of beforehand allows the employer to make efficient use of the available information at each step and reduce the total expense incurred during the labeling process.

In the present paper, we consider a typical labeling scenario in which the easiness of the objects and the quality of the workers are both heterogeneous. We focus on a dynamic environment, where workers arrive over time while the employer is running the task, so that incoming workers can be assigned to individual objects dynamically. To harness the potential of this dynamic decision system, we propose several heuristic-based label allocation strategies that adaptively choose which object to label next, based on the algorithmic estimates of object and worker quality from all the labels obtained so far. Using experiments on both synthetic and real-world data sets, we demonstrate that our proposed dynamic label allocation methods can achieve significant savings in labeling expenses and completion time.

Another contribution of this paper is to use a decision-theoretic approach to generate two performance metrics for each worker, both of which allow the employer to separate correctable errors from uncorrectable errors that workers make and take into account the costs of different classification errors. In particular, the worker *value* metric directly measures the monetary contribution of each label provided by a worker towards meeting the quality requirements of the employer and provides a basis for the employer to grant monetary bonuses to high-quality workers who contribute more than they earn and block inferior workers whose contributed value is not worth the payment.

Our paper responds to the call for more research on design science in IS field (e.g., Hevner et al., 2004; March and Storey, 2008; Kuechler and Vaishnavi, 2012; Gregor and Hevner, 2013; Goes, 2014). As mentioned earlier, companies today invest substantial amounts of money in gathering information and knowledge from crowdsourced workers. Therefore, the problem of cost reduction in information acquisition is of tremendous importance to business organizations. By formulating a decision problem in the dynamic crowd labeling environment, developing techniques that can manage quality assurance cost-effectively, and demonstrating the efficacy of the proposed methods via rigorous experimentation, this work will help companies to dramatically reduce data acquisition costs and engage in faster and more efficient decision-making in their business processes.

The remainder of the paper is organized as follows. Section 2 reviews the related literature. Section 3 outlines the modeling assumptions and formalizes the problem. Section 4 describes the inference algorithms for estimating the true classes of objects and the quality of workers. Section 5 proposes several heuristic-based dynamic label allocation strategies, which aim to reduce labeling expenses while maintaining the required level of data quality. Section 6 and Section 7 evaluate the performance of inference and allocation algorithms using simulated and real-world datasets, respectively. Section 8 introduces two scalar metrics for evaluating the performance of heterogeneous workers and discusses the potential of worker value metric for compensation scheme design. Section 9 concludes by presenting practical implications, limitations, and directions for future research.

2 Literature Review

In this section, we survey the relevant literature in three streams of research: quality estimation and control, worker performance metric, and active information acquisition.

2.1 Quality Estimation and Control

A simple approach to measure the quality of submitted answers is to use *gold* data: insert a small percentage of tasks for which the correct answers are known, and measure the performance on these tasks. The testing of worker quality using gold labels is related to, but distinct from, two lines of research: test theory in psychometrics and education (Crocker and Algina, 2006; DeMars, 2010), and acceptance sampling in operation management (Dodge, 1973; Wetherill and Chiu, 1975; Berger, 1982; Schilling, 1982). Existing test theory models do not consider the additional costs incurred in labeling gold data, which is analogous to the inspection cost in manufacturing process. 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 crowd labeling markets that deal with

information goods, low-quality work can be combined to provide high-quality outcomes.

Another method for ensuring quality is to ask multiple workers to complete the same task and use majority voting (MV) to identify the correct answer. In reality, most employers check labels provided by workers with majority voting and dismiss workers systematically in disagreement with the majority. This approach has two undesirable properties: first, it does not account for heterogeneity in the exhibited quality of workers; and second, it suffers in the face of diligent and informative workers whose answers are wrong but correctable.

Several more advanced aggregation methods have been developed in the past years. Dawid and Skene (1979) present an expectation maximization (EM) algorithm to simultaneously estimate the true responses for patients and the error rates of observers. The algorithm iterates until convergence, following two steps: (1) estimates the true response for each patient, using records given by all observers, accounting for the error rates of each observer; and (2) estimates the error rates of each observer by comparing the submitted records with estimated true responses. Variations of the algorithm were recently proposed by Carpenter (2008) and by Raykar et al. (2010). Welinder et al. (2010) develop a generative Bayesian model in which each annotator is a multidimensional entity with variables representing competence, expertise and bias. Inspired by the standard belief propagation algorithm, Karger et al. (2011) introduce a novel message-passing technique to jointly infer the correct answers of the tasks and the reliability of workers. Whitehill et al. (2009) incorporate task difficulty into the labeling process and present a probabilistic model that simultaneously infers the expertise of each worker, and the label and the difficulty of each task. The decision systems in these papers are all static and involve no adaptive allocation of labels.

2.2 Worker Performance Metric

All the above inference algorithms generate some indicators of worker performance in scalar, vector or matrix form. For example, MV measures the accuracy rate of each worker by the proportion of the labels submitted by the worker in agreement with the majority labels. The EM algorithm proposed by Dawid and Skene (1979) returns a *confusion matrix* which lists the probabilities of different classification errors made by each worker. Welinder et al. (2010) measure worker ability in a multidimensional space with each element modeling the worker's individual weighting on each of the major components of the annotation task. Karger et al. (2011) use a set of task-specific worker messages to represent the belief of how reliable a worker is in labeling each specific task. Whitehill et al. (2009) employ a scalar value to model the expertise of each worker.

However, none of these metrics can effectively quantify the contributed value of each label provided by an individual worker in meeting the quality assurance needs of the employer. First, they cannot separate correctable errors from uncorrectable errors that workers make. For example, a malicious worker may always

submit wrong labels, but these labels are informative as they can be reversed to uncover the truth. In such cases, the naive measurement of accuracy rate results in underestimates of the value of workers who consistently give predictably incorrect answers. Second, they fail to take into account the relative costs of different types of classification errors. Understandably, some types of misclassification errors incur significantly higher costs than others. For example, allowing a porn image to pass a moderation filter is often more costly than blocking a legitimate image incorrectly. Third, a number of workers often need to work in tandem to generate labels of acceptable quality, therefore, it is more appropriate to evaluate the performance of each worker in a multiple-label setting than treating them as isolated. In our work, we propose a value-based performance metric that directly measures the monetary contribution of each worker in a multiple-label setting, after fixing correctable worker errors and accounting for heterogeneity in misclassification costs across different types of classification errors.

2.3 Active Information Acquisition

Active information acquisition, which focuses on gathering various types of information incrementally, so as to achieve different objectives cost-effectively, has been an important topic in machine learning and management literature. Moore and Whinston (1986, 1987) develop a theoretical decision-making framework in which the decision-maker gathers costly information optimally and sequentially to reduce the uncertainty associated with the final decisions. There have been a large number of papers devoted to active learning (e.g., Cohn et al., 1994; Lewis and Gale, 1994; Roy and McCallum, 2001; Saar-Tsechansky and Provost, 2004), which aim to economize resources on training instances that are more likely to be informative for building classifiers. Another stream of papers (Lizotte et al., 2002; Zheng and Padmanabhan, 2006; Saar-Tsechansky and Provost, 2007; Saar-Tsechansky et al., 2009) study the active feature-value acquisition problem in scenarios where the feature values of the training data are costly to acquire.

In the context of dynamic label allocation using multiple noisy workers, Sheng et al. (2008) and Ipeirotis et al. (2014) develop several selective repeated-labeling strategies and show that selective allocation of labeling resources can improve the overall labeling quality and model prediction accuracy. However, both papers assume that all workers have equal level of quality when labeling the same instance and the costs incurred by different classification errors are identical, which rarely hold in real-life scenarios.

Another emerging set of research papers take an additional step by allowing employers to selectively target workers when requesting labels. The underlying assumption is that employers can arbitrarily exploit high-quality workers by asking them to label as many objects as possible. For instance, Welinder and Perona (2010) propose a dynamic allocation approach in which the employer can prioritize expert workers by asking them to label more in the annotation process. Chen et al. (2014) formulate a budget allocation problem

where the employer can simultaneously choose which instance to label next and which worker to assign the task to. These studies have limited applicability in platforms where workers' arrival is exogenous and not under the control of employers. For example, workers on AMT arrive in real time and may work for as long or as little as they wish, depending on their own interests and time constraints. In the present paper, we work under the assumption that workers' arrival and participation are not controlled by the employer, and focus on the dynamic task allocation problem that aims to reduce labeling expenses by adaptively selecting which object to assign to a worker conditional on her agreement to continue providing labels.

3 Modeling Framework

In this section, we describe our modeling assumptions and formalize the problem. Table 1 summarizes the notation used in the paper.

Notation	Definition
$t^{(o)}$	The true class of object (o)
$\mathbf{c}$	The misclassification cost matrix
c_{ij}	The cost incurred when an object with true class i is classified into class j
τ_c	The threshold for the average misclassification cost
V	The value of each object with average misclassification cost below τ_c
$l_{(o)}^{(k)}$	The label that worker (k) assigns to object (o)
L	The set of observed labels $\{l_{(o)}^{(k)}\}$
$\boldsymbol{\alpha}^{(k)}$	The quality vector of worker (k)
$\alpha_i^{(k)}$	The quality of worker (k) on labeling objects in class i
$\beta^{(o)}$	The easiness of object (o)
$\hat{t}^{(o)}$	The estimated true class of object (o)
$K^{(o)}$	The set of workers who assign labels to object (o)
$O^{(k)}$	The set of objects labeled by worker (k)
$I(\cdot)$	The indicator function for an event
$\lvert\cdot\rvert$	The cardinality of a set
$\mathbf{p}^{(o)}$	The vector with probability estimates for the true class of object (o)
$p_i^{(o)}$	The estimated probability that the true class of object (o) is i
$q^{(k)}$	Accuracy rate of worker (k) (for MV)
$\boldsymbol{\pi}$	The vector with prior probabilities of all classes
π_i	The prior probability of class i
$\mathbf{e}^{(k)}$	The confusion matrix for worker (k) (for EM)
$e_{ij}^{(k)}$	The probability that worker (k) labels an object of class i into class j (for EM)
$L^{(o)}$	The set of observed labels on object (o)
$\hat{\pi}_j^{(k)}$	The estimated prior probability that worker (k) assigns label j

Table 1: Notation used in the paper

3.1 Scenario

We consider a typical scenario in crowd labeling. An employer has a set of unlabeled objects and wants them to be labeled with the correct classes (e.g., judging whether a Facebook post contains hate speech). The employer may incur different costs in making different types of classification errors. For instance, it is more costly for Facebook to classify a hate speech post as okay than to classify a legitimate post as hate speech. We assume that the employer's misclassification costs can be represented by a matrix $\mathbf{c}$: the cost c_{ij} is incurred when an object with true class i is categorized into class j.[6] The average misclassification cost is used to quantify the quality of labeling. The goal of the employer is to guarantee that the average misclassification cost will not exceed a threshold τ_c. We further assume that the employer can derive a value of V from each labeled object with average misclassification cost not exceeding τ_c.

The employer posts the task on a micro-crowdsourcing platform (e.g., AMT, CrowdFlower, CrowdSource). Workers arrive at the platform over time and search for tasks they are interested in. When a worker agrees to perform the task, the employer presents the to-be-labeled objects to the worker one after another, until she stops working or the employer achieves the quality requirements.

3.2 The Labeling Model

In the labeling task, each object (o) is associated with a *latent* true class $t^{(o)}$, picked from one of two possible classes 0 or 1 (e.g., positive/negative, useful/not useful). The true class $t^{(o)}$ is unknown and the task is to infer the true class for each object (o). The objects to be labeled may vary in their level of easiness. For example, a hate speech that directly attacks people based on their ethnicity is easier to identify than a hate speech that demeans people in a subtle way.

To incorporate the effect of object easiness, we adapt the labeling model from Whitehill et al. (2009), but allowing workers' quality to vary across the two classes. The observed label $l_{(o)}^{(k)}$ provided by worker (k) on object (o) is jointly determined by three factors: (1) the quality of worker (k); (2) the easiness of object (o); and (3) the true class of object (o).

We model the quality of each worker (k) by a two-dimensional vector $\boldsymbol{\alpha}^{(k)} = (\alpha_0^{(k)}, \alpha_1^{(k)})$, where $\alpha_i^{(k)} \in (+\infty, -\infty)$ represents worker (k)'s quality on labeling objects belonging to class i. Here, $\alpha_i^{(k)} = +\infty$ means worker (k) always labels objects in class i correctly; and $\alpha_i^{(k)} = -\infty$ means worker (k) always labels objects in class i incorrectly. Note that unlike Whitehill et al. (2009), we don't impose the constraint that $\alpha_0^{(k)} = \alpha_1^{(k)}$ and allow the quality of each worker to vary by classes. For example, if worker (k) labels all objects into class i, then $\alpha_i^{(k)} = +\infty$ and $\alpha_{1-i}^{(k)} = -\infty$.

The easiness of each object (o) is modeled by $\beta^{(o)}$, where $\beta^{(o)} \in [0, +\infty)$ is constrained to be positive.

[6] By definition, $c_{ij} = 0$ when $i = j$.

Here, $\beta^{(o)} = 0$ means object (o) is very difficult to label and even a high-quality worker only has a half chance of labeling it correctly; and $\beta^{(o)} = +\infty$ means object (o) is very easy to label and even a low-quality worker can label it correctly with 100% probability.

Under the labeling model, the log adds of the obtained label being correct is a bilinear function of the quality of worker (k) on class $t^{(o)}$ and the easiness of the object (o), i.e.,

$$\log \frac{p\big(l_{(o)}^{(k)} = t^{(o)} | \alpha_{t^{(o)}}^{(k)}, \beta^{(o)}\big)}{1 - p\big(l_{(o)}^{(k)} = t^{(o)} | \alpha_{t^{(o)}}^{(k)}, \beta^{(o)}\big)} = \alpha_{t^{(o)}}^{(k)} \beta^{(o)}$$

Thus, the label $l_{(o)}^{(k)}$ given by worker (k) to object (o) is generated as follows:

$$p\big(l_{(o)}^{(k)} = t^{(o)} | \alpha_{t^{(o)}}^{(k)}, \beta^{(o)}\big) = \frac{1}{1 + e^{-\alpha_{t^{(o)}}^{(k)} \beta^{(o)}}} \tag{1}$$

and

$$p\big(l_{(o)}^{(k)} = 1 - t^{(o)} | \alpha_{t^{(o)}}^{(k)}, \beta^{(o)}\big) = 1 - \frac{1}{1 + e^{-\alpha_{t^{(o)}}^{(k)} \beta^{(o)}}} = \frac{1}{1 + e^{\alpha_{t^{(o)}}^{(k)} \beta^{(o)}}} \tag{2}$$

Understandably, the probability that worker (k) labels object (o) correctly increases with her labeling quality on class $t^{(o)}$ and the easiness of the object (o).

4 Inference

In this section, we describe several algorithms for inferring the true classes of objects and the quality of workers.

4.1 Majority Voting (MV)

The simplest method to estimate the true class of an object is majority voting (MV), which simply ignores any heterogeneity in worker quality and takes the majority label provided by multiple workers. The performance of each worker is measured by accuracy rate (i.e., how frequently the worker agrees with the majority label). Algorithm 1 presents a sketch of the MV algorithm. Due to its simplicity, MV is commonly used by employers who lack competence in data processing.

4.2 Message Passing (MP)

Inspired by the standard belief propagation algorithm, Karger et al. (2011) introduce a message-passing (MP) algorithm which jointly infers the true classes of objects and the reliability of workers. The algorithm iteratively operates on a set of object messages and worker messages: at each object update, it gives more

Input: The set of observed labels $L = \{l_{(o)}^{(k)}\}$

Output: Estimated true class $\hat{t}^{(o)}$ for each object (o), accuracy rate $q^{(k)}$ for each worker (k)

Estimate the class probability estimates for each object (o): $p_i^{(o)} = \frac{\sum_{(k)\in K^{(o)}} I\left(l_{(o)}^{(k)}=i\right)}{|K^{(o)}|}$;

Estimate the true class using the majority label for object (o): $\hat{t}^{(o)} = \arg\max_{i\in\{0,1\}} p_i^{(o)}$;

Estimate the accuracy rate of each worker (k): $q^{(k)} = \frac{\sum_{(o)\in O^{(k)}} I\left(l_{(o)}^{(k)}=\hat{t}^{(o)}\right)}{|O^{(k)}|}$.

Algorithm 1: Majority voting (MV) inference algorithm

Input: The set of observed labels $L = \{l_{(o)}^{(k)}\}$

Output: Estimated true class $\hat{t}^{(o)}$ for each object (o), object message $\{x_{(o)\to(k)}\}$ and worker message $\{y_{(k)\to(o)}\}$ for each object (o) and worker (k) with $l_{(o)}^{(k)} \in L$

Initialize each worker message: draw $y_{(k)\to(o)}$ from a Gaussian distribution $\mathcal{N}(1,1)$;

while *not converged* **do**

  Update each object message: $x_{(o)\to(k)} = \sum_{(k')\in K^{(o)}\setminus(k)} (2l_{(o)}^{(k)} - 1) y_{(k')\to(o)}$;

  Update each worker message: $y_{(k)\to(o)} = \sum_{(o')\in O^{(k)}\setminus(o)} (2l_{(o)}^{(k)} - 1) x_{(o')\to(k)}$;

end

Calculate the estimated true class for each object (o): $\hat{t}^{(o)} = \frac{1}{2}\left(1 + \text{sign}\left(\sum_{(k)\in K^{(o)}} (2l_{(o)}^{(k)} - 1) y_{(k)\to(o)}\right)\right)$.

Algorithm 2: Message passing (MP) inference algorithm

weight to labels that come from more trustworthy workers; and at each worker update, it adds more confidence in that worker if the labels she gives on other objects agree with the current estimates of object labels. The details of the MP algorithm are given Algorithm 2.[7]

4.3 Expectation Maximization (EM)

Another advanced inference technique is expectation maximization (EM), first proposed by Dawid and Skene (1979) in the context of medical diagnosis. The algorithm iterates until convergence, following two steps: (1) estimates the true class for each object using the labels provided by a set of workers, accounting for the error rates of each worker; and (2) estimates the error rates of each worker by comparing the submitted labels with estimated true class for each object. The performance of each worker (k) is represented by a *confusion matrix* $\mathbf{e}^{(k)}$, where $e_{ij}^{(k)}$ gives the probability that worker (k) classifies an object of class i into class j.

To incorporate priors on worker quality, we move from maximum likelihood estimates to Bayesian ones. If the true class of an object is i, we model the error rates of the worker (k) as a Beta distribution with parameter vector $\boldsymbol{\theta}_i^{(k)}$. The value of $\theta_{ij}^{(k)}$ is given by $\theta_{ij}^{(k)} = \lambda_{ij}^{(k)} + n_{ij}^{(k)}$, where $n_{ij}^{(k)}$ represents the number of times that the worker classifies objects of class i into class j and $\lambda_{ij}^{(k)}$ captures the prior belief. Using this

[7] The algorithm is slightly different from the original one presented in Karger et al. (2011) since the possible label set is now $\{0,1\}$ instead of $\{-1,1\}$.

Input: The set of observed labels $L = \{l_{(o)}^{(k)}\}$, priors $\boldsymbol{\lambda}^{(k)}$

Output: Class probability estimates $\mathbf{p}^{(o)}$ for each object (o), confusion matrix $\mathbf{e}^{(k)}$ for each worker (k), class prior estimates $\hat{\boldsymbol{\pi}}$

Initialize class probability estimates for each object (o): $p_i^{(o)} = \frac{\sum_{(k)\in K^{(o)}} I\left(l_{(o)}^{(k)}=i\right)}{|K^{(o)}|}$;

while *not converged* **do**

  Estimate the $\boldsymbol{\theta}^{(k)}$: $\theta_{ij}^{(k)} = \lambda_{ij}^{(k)} + n_{ij}^{(k)} = \lambda_{ij}^{(k)} + \sum_{(o)\in O^{(k)}} p_i^{(o)} I\left(l_{(o)}^{(k)} = j\right)$;

  Estimate the confusion matrix $\mathbf{e}^{(k)}$: $e_{ij}^{(k)} = \frac{\theta_{ij}^{(k)}}{\sum_q \theta_{iq}^{(k)}}$;

  Estimate the class priors: $\hat{\pi}_i = \frac{\sum_{(o)} p_i^{(o)}}{|O|}$;

  Compute the class probability estimates for each object (o):

$$p_i^{(o)} = \frac{\hat{\pi}_i \prod_{(k)\in K^{(o)}} \prod_m (e_{im}^{(k)})^{I\left(l_{(o)}^{(k)}=m\right)}}{\sum_q \hat{\pi}_q \prod_{(k)\in K^{(o)}} \prod_m (e_{qm}^{(k)})^{I\left(l_{(o)}^{(k)}=m\right)}};$$

end

Algorithm 3: Bayesian expectation maximization (EM) inference algorithm

strategy, the error rates of a worker can be fully captured by two Beta distributions. Algorithm 3 presents a sketch of the process, where $\boldsymbol{\theta}^{(k)}$ parameterizes the error rate distributions of worker (k) and $\mathbf{e}^{(k)}$ is defined by the expected values.

4.4 Generative Model of Labels, Abilities, and Difficulties (GLAD)

All the previous inference algorithms ignore the possible heterogeneity of object easiness and simply attribute the generation of noisy labels to imperfect worker quality. But as illustrated in Section 3.2, the observed labels provided by workers on a particular object may also depend on the easiness of the object.

Following Whitehill et al. (2009), we use an expectation-maximization (EM) approach to obtain the maximum likelihood estimates of the $\boldsymbol{\alpha}^{(k)}$, $\beta^{(o)}$, and $t^{(o)}$ for each worker (k) and each object (o).

E-step: The posterior probability of $t^{(o)}$ given $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\beta^{(o)}\}$ is characterized by:

$$\begin{aligned}
p\left(t^{(o)}|L,\{\boldsymbol{\alpha}^{(k)}\},\{\beta^{(o)}\}\right) &= p\left(t^{(o)}|L^{(o)},\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\},\beta^{(o)}\right) \\
&\propto p\left(t^{(o)}|\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\},\beta^{(o)}\right) p\left(L^{(o)}|t^{(o)},\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\},\beta^{(o)}\right) \\
&\text{since } l_{(o)}^{(k)}\text{'s are cond. indep. given } t^{(o)},\{\boldsymbol{\alpha}^{(k)}\} \text{ and } \beta^{(o)} \\
&\propto p(t^{(o)}) \prod_{(k)\in K^{(o)}} p\left(l_{(o)}^{(k)}|t^{(o)},\alpha_{t^{(o)}}^{(k)},\beta^{(o)}\right) \\
&\propto p(t^{(o)}) \prod_{(k)\in K^{(o)}} \left(\frac{1}{1+e^{-\alpha_{t^{(o)}}^{(k)}\beta^{(o)}}}\right)^{I\left(l_{(o)}^{(k)}=t^{(o)}\right)} \left(\frac{1}{1+e^{\alpha_{t^{(o)}}^{(k)}\beta^{(o)}}}\right)^{I\left(l_{(o)}^{(k)}=1-t^{(o)}\right)}
\end{aligned} \tag{3}$$

Following equation (3), we can calculate the posterior probability of $t^{(o)}$ using the prior probability of

$t^{(o)}$, the values of $\{\boldsymbol{\alpha}^{(k)}|(k) \in K^{(o)}\}$ and the value of $\beta^{(o)}$ estimated from the previous M-step.

M-step: We maximize the auxiliary function Q, which is defined as the expectation of the joint log-likelihood of the observed and hidden variables $(L, \{t^{(o)}\})$ given the parameters $(\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\})$, where the values of hidden variables $\{t^{(o)}\}$ are computed during the previous E-step. We can also impose a prior on each parameter. The prior probabilities of $\alpha_0^{(k)}$, $\alpha_1^{(k)}$, and $\beta^{(o)}$ are denoted as $p(\alpha_0^{(k)})$, $p(\alpha_1^{(k)})$, and $p(\beta^{(o)})$, respectively.

$$
\begin{aligned}
Q(\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\}) &= \mathbb{E}\big[\ln\big(p(L, \{t^{(o)}\}|\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\})p(\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\})\big)\big] \\
&= \mathbb{E}\big[\ln \prod_{(o)} \big(p(t^{(o)}) \prod_{(k)\in K^{(o)}} p(l_{(o)}^{(k)}|t^{(o)}, \alpha_{t^{(o)}}^{(k)}, \beta^{(o)})\big)\big] \\
&\quad + \ln \prod_{(k)} \prod_{i=0}^{1} p(\alpha_i^{(k)}) + \ln \prod_{(o)} p(\beta^{(o)}) \\
&= \sum_{(o)} \mathbb{E}\big[\ln p(t^{(o)})\big] + \sum_{(o)} \sum_{(k)\in K^{(o)}} \mathbb{E}\big[\ln p(l_{(o)}^{(k)}|t^{(o)}, \alpha_{t^{(o)}}^{(k)}, \beta^{(o)})\big] \\
&\quad + \sum_{(k)} \sum_{i=0}^{1} \ln p(\alpha_i^{(k)}) + \sum_{(o)} \ln p(\beta^{(o)})
\end{aligned}
\tag{4}
$$

where the expectation is taken over $\{t^{(o)}\}$ estimated during the previous E-step. The values of $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\beta^{(o)}\}$ are obtained by maximizing the auxiliary function Q. This is not directly solvable, therefore we apply a gradient ascent approach to find parameter values that locally maximize Q (The details of the gradient ascent approach are provided in Appendix B).

We present a sketch of the inference process in Algorithm 4. Note that when we assume that all the objects are equally difficult (i.e., $\beta^{(o)} = \beta$ where β is a constant), GLAD degenerates to EM with the following relationship: $e_{ii}^{(k)} = \dfrac{1}{1 + e^{-\alpha_i^{(k)}\beta}}$ and $e_{i,1-i}^{(k)} = \dfrac{1}{1 + e^{\alpha_i^{(k)}\beta}}$.

4.5 Estimated and Actual Misclassification Cost

In the four inference algorithms presented above, MV and MP return the estimated true class $\hat{t}^{(o)}$ for each object (o), while EM and GLAD return the class probability estimates $\mathbf{p}^{(o)}$ for each object (o). If the class probability estimates of the object are available, we can calculate the *estimated misclassification cost* as follows:

Proposition 1 *Given the misclassification cost matrix* $\mathbf{c}$ *and the class probability estimates* $\mathbf{p}^{(o)}$ *for object* (o)*, the estimated misclassification cost of object* (o) *is* $EstCost(\mathbf{p}^{(o)}) = \min_{j\in\{0,1\}} \sum_{i=0}^{1} p_i^{(o)} c_{ij}$.

The estimated misclassification cost if we report j as the true class is equal to the posterior probability of the object (o) belonging to class i (namely, $p_i^{(o)}$) multiplied with the associated cost of classifying an object

Input: The set of observed labels $L = \{l_{(o)}^{(k)}\}$, priors $p(\alpha_0^{(k)})$, $p(\alpha_1^{(k)})$, and $p(\beta^{(o)})$

Output: Class probability estimates $\mathbf{p}^{(o)}$ for each object (o), easiness $\hat{\beta}^{(o)}$ of each object (o), quality vector $\hat{\boldsymbol{\alpha}}^{(k)}$ for each worker (k), class prior estimates $\hat{\boldsymbol{\pi}}$

Initialize the easiness estimate for each object (o): $\hat{\beta}^{(o)} = 1$;

Initialize class probability estimates for each object (o): $p_i^{(o)} = \frac{\sum_{(k)\in K^{(o)}} I\left(l_{(o)}^{(k)}=i\right)}{|K^{(o)}|}$;

Estimate the class priors: $\hat{\pi}_i = \frac{\sum_{(o)} p_i^{(o)}}{|O|}$;

while *not converged* **do**

  Obtain the estimated quality vector $\hat{\boldsymbol{\alpha}}^{(k)}$ for each worker (k) and the estimated easiness $\hat{\beta}^{(o)}$ of each object (o) by maximizing the auxiliary function $Q(\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\})$ in Equation (4) using gradient ascent approach;

  Compute the class probability estimates for each object (o):

$$p_i^{(o)} = \frac{\hat{\pi}_i \prod_{(k)\in K^{(o)}} \left(\frac{1}{1+e^{-\hat{\alpha}_i^{(k)}\hat{\beta}^{(o)}}}\right)^{I\left(l_{(o)}^{(k)}=i\right)} \left(\frac{1}{1+e^{\hat{\alpha}_i^{(k)}\hat{\beta}^{(o)}}}\right)^{I\left(l_{(o)}^{(k)}=1-i\right)}}{\sum_q \hat{\pi}_q \prod_{(k)\in K^{(o)}} \left(\frac{1}{1+e^{-\hat{\alpha}_q^{(k)}\hat{\beta}^{(o)}}}\right)^{I\left(l_{(o)}^{(k)}=q\right)} \left(\frac{1}{1+e^{\hat{\alpha}_q^{(k)}\hat{\beta}^{(o)}}}\right)^{I\left(l_{(o)}^{(k)}=1-q\right)}};$$

  Estimate the class priors: $\hat{\pi}_i = \frac{\sum_{(o)} p_i^{(o)}}{|O|}$;

end

Algorithm 4: GLAD expectation maximization inference algorithm

of class i into class j (namely, c_{ij}), aggregated over all possible i's. Clearly, the best decision is to report the class that incurs the minimum cost. Therefore, the reported class label for EM and GLAD is given by $\hat{t}^{(o)} = \arg\min_{j\in\{0,1\}} \sum_{i=0}^{1} p_i^{(o)} c_{ij}$. Notice that unlike MV and MP, the object class label reported by EM and GLAD might vary depending on the misclassification cost matrix.

If the true class of the object is known, we can also calculate the *actual misclassification cost*:

Proposition 2 *Given the misclassification cost matrix* $\mathbf{c}$*, the true class label* $t^{(o)}$ *and the estimated class label* $\hat{t}^{(o)}$ *for object* (o)*, the actual misclassification cost of object* (o) *is* $ActualCost(\hat{t}^{(o)}) = c_{t^{(o)}\hat{t}^{(o)}}$.

5 Dynamic Label Allocation

In the previous section, we focus on a static setting: given all the labels provided by workers, we use several different approaches to infer object class and worker quality. In real crowdsourcing marketplaces, labels are often obtained incrementally and dynamically, therefore the $(n+1)$-th label allocation decision can be made based on the n labels collected so far. Intuitively, it is preferable for the employer to prioritize labels to objects that are more likely to achieve a greater reduction in misclassification cost. The challenge facing the employer is to devise a label allocation strategy that minimizes the number of labels required to achieve a certain level of data quality (measured by average misclassification cost), or equivalently, minimizes the average misclassification cost with a given number of labels.

To find the optimal allocation strategy, the employer needs to solve the following optimization problem:

$$\underset{z}{\text{minimize}} \quad \mathbb{E}_z\Big[\sum_{(o)} ActualCost(\hat{t}^{(o)}|_N)\Big] \tag{5}$$

where $|_N$ denotes the estimates at the final step N. $\mathbb{E}_z$ represents the expectation taken over the sample paths $\{(o_1), (o_2), \cdots, (o_N)\}$ generated by an allocation strategy z.

The optimization problem in (5) is a finite horizon multi-armed bandit (MAB) problem, where each object corresponds to an arm, while pulling an arm is equivalent to assigning the next label to a particular object. However, our problem is more challenging because the rewards can only be realized at the final step when the average misclassification cost does not exceed a threshold and so the intermediate rewards at each step are not deducible or distinguishable. This problem is computationally intractable, therefore we resort to heuristic approaches to find approximate solutions.

Below, we propose several heuristic-based dynamic label allocation strategies with the aim of reducing the label resources required to achieve the desired data quality.

5.1 Message Passing-Reliability (MP-Reliab)

The first strategy is motivated by the MP algorithm in Section 4.2, which estimates the label for each object based on the sign of a weighted sum of the answers provided by workers. While the predicted label is binary (i.e., 0 or 1), the weighted sum is a real value. The further away the sum is from zero, the more reliable the prediction is. Therefore, it makes sense to allocate more labels to objects whose weighted sums, based on the existing labels, are closer to the decision threshold zero.

To formalize this idea, we define the following heuristic function:

$$h^{(o)}_{MP\text{-}Reliab} = -\left|\sum_{(k)\in K^{(o)}} (2l^{(k)}_{(o)} - 1) y_{(k)\to(o)})\right|$$

Here, all the notations are from Section 4.2. The negative sign is introduced to get a larger function value when the deviation from zero is smaller.

5.2 Expectation Maximization-Cost (EM-Cost) and GLAD-Cost

One drawback of the MP-Reliab strategy is that it cannot incorporate the heterogeneous costs of different classification errors into the allocation decision. Fortunately, the EM and GLAD inference algorithms presented in Section 4.3 and 4.4 are able to generate class probability estimates for all the objects, which can then be utilized to help the employer make a more informed decision.

The estimated misclassification cost of the object is important to inform the allocation decision on which object to assign the next label to. Based on Proposition 1, the highest cost is incurred when the two different label predictions yield the same misclassification cost (i.e., $p_0^{(o)}c_{00} + p_1^{(o)}c_{10} = p_0^{(o)}c_{01} + p_1^{(o)}c_{11}$). When the estimated cost is high (i.e., the costs of two label predictions are similar), a small variation in class probability estimates can lead to totally different label predictions. On the other hand, when the estimated cost is low, the same variation might not cause any change in label prediction. Therefore, the average misclassification cost is more likely to be reduced when the additional labels are allocated to objects with higher estimated costs.

The heuristic functions based on this idea are:

$$h^{(o)}_{EM\text{-}Cost} = EstCost(\mathbf{p}^{(o)}_{EM}) \quad \text{and} \quad h^{(o)}_{GLAD\text{-}Cost} = EstCost(\mathbf{p}^{(o)}_{GLAD})$$

where $\mathbf{p}^{(o)}_{EM}$ represent the class probability estimates for object (o) using EM inference algorithm, and $\mathbf{p}^{(o)}_{GLAD}$ are the class probability estimates for object (o) returned by GLAD algorithm.

5.3 GLAD-Cost-Variation (GLAD-CostV)

One criticism of the two cost-based approaches is that labeling the object with the highest estimated misclassification cost does not necessarily lead to the greatest cost reduction. For example, a difficult object is more likely to have high estimated cost; however, the cost reduction induced by an additional label may be marginal as even a high-quality worker has a considerable chance of providing an incorrect label.

To remedy this deficiency, we want to incorporate the expected reduction in estimated misclassification cost induced by one additional label into the heuristic function. However, as shown in Proposition 3, the estimated misclassification cost of an object is very likely to stay the same in expectation.

Proposition 3 *Assume that the easiness of an object (o) is $\beta^{(o)}$ and its class probability estimate is $\mathbf{p}^{(o)}|_m$ after querying m workers, and now there arrives a worker (k) with quality vector $\boldsymbol{\alpha}^{(k)}$. The estimated misclassification cost will stay the same in expectation if the predicted label does not change with the adding of the $(m+1)$-th label by worker (k), i.e., $EstCost^{(o)}|_m = \mathbb{E}(EstCost^{(o)}|_{(m+1)})$.*

Proof: Proof. See Appendix C.1. □

In practice, the same predicted label condition in Proposition 3 can be easily met. When the predicted label of object (o) at step m is in agreement with the new label provided by worker (k), the model predicts the same label at step $(m+1)$; when the two labels conflict with each other, the predicted label at step $(m+1)$ still does not change as long as the model has more confidence in the collective label of the first m workers than in the label of the $(m+1)$-th worker. Therefore, we cannot use the expected reduction

in estimated misclassification cost $EstCost^{(o)}|_m - \mathbb{E}(EstCost^{(o)}|_{(m+1)})$ as the heuristic function to solve the optimization problem.

As an alternative, we propose a new approach that selects the next object to label based on the expected variation in estimated misclassification cost. The underlying intuition is that the addition of one more label leads to higher cost variation for more uncertain objects. If the model is confident about the true class of a particular object, an additional label would result in little cost reduction when it agrees with the previous label prediction, and little cost increment when it disagrees with the previous prediction. The heuristic function is:

$$\begin{aligned} h^{(o)}_{GLAD\text{-}CostV} &= \mathbb{E}(\Big|EstCost^{(o)}|_m - EstCost^{(o)}|_{(m+1)}\Big|) \\ &= p(l^{(k)}_{(o)} = 0)\Big|EstCost^{(o)}|_m - EstCost^{(o)}|^0_{(m+1)}\Big| + p(l^{(k)}_{(o)} = 1)\Big|EstCost^{(o)}|_m - EstCost^{(o)}|^1_{(m+1)}\Big| \end{aligned}$$

where $EstCost^{(o)}|^0_{(m+1)}$ represents the estimated misclassification cost of the object when the additional label provided by the worker equals to 0, and $EstCost^{(o)}|^1_{(m+1)}$ represents the estimated misclassification cost of the object with the additional label being 1.

Note that this cost variation approach cannot help when the inference algorithm employed is EM, which is due to the basic assumption of EM model that the same worker has equal likelihood of making errors on all objects, regardless of the difficulty of each object.

Proposition 4 *The expected variation in estimated misclassification cost of an object* (o) *under EM algorithm only depends on the class probability estimate* $\mathbf{p}^{(o)}|_m$*, the confusion matrix* $\mathbf{e}^{(k)}$ *of the worker* (k) *who provides the next label, and the cost matrix* $\mathbf{c}$*.*

Proof: Proof. See Appendix C.2. □

Based on Proposition 4, we can claim that if two objects have the same class probability estimates (and thus the same estimated misclassification cost), assigning the next label to either of them will produce the same expected variation in estimated misclassification cost.

All the four strategies proposed above prioritize labels based on different heuristics. We present a general framework for dynamic label allocation in Algorithm 5. Note that the inference algorithms used for deriving the values of heuristic functions can be different from the inference algorithms used for estimating misclassification cost. For instance, we may use MP-Reliab (which relies on MP to infer $h^{(o)}_{MP\text{-}Reliab}$) for allocating labels and use EM for estimating misclassification cost.

5.4 Batch Processing

Two minor points limit the applicability of the dynamic allocation strategies described above in real-world large data environments. First, at each time point, we need to compute the values of the heuristic functions

Input: The set of objects $O = \{(o)\}$ to be labeled, misclassification cost matrix $\mathbf{c}$, cost threshold τ_c, heuristic function $h^{(o)} \in \{h^{(o)}_{MP\text{-}Reliab}, h^{(o)}_{EM\text{-}Cost}, h^{(o)}_{GLAD\text{-}Cost}, h^{(o)}_{GLAD\text{-}CostV}\}$, inference and cost estimation algorithm $\eta \in \{\text{EM, GLAD}\}$

Output: Predicted class label $\hat{t}^{(o)}$ for each object (o)

Initialize $L = \varnothing$, $O^{(k)} = \varnothing$ for each worker (k), $avg\_cost = \infty$;
while $avg\_cost > \tau_c$ **do**
  When a worker (k) is ready to accept the next task,
    Select the next object to label according to $(o) = \arg\max_{(o') \in O \setminus O^{(k)}} h^{(o')}$;
  Once worker (k) finishes the task,
    Add the acquired label $l^{(k)}_{(o)}$ to the label set: $L = L + l^{(k)}_{(o)}$;
    Add object (o) to the set of objects labeled by worker (k): $O^{(k)} = O^{(k)} + (o)$;
  Using algorithm η, estimate the class probability estimates $\mathbf{p}^{(o)}$ for each object (o);
  $sum\_cost = 0$;
  for $(o) \in O$ **do**
    Calculate the predicated label $\hat{t}^{(o)}$ and the estimated misclassification cost $EstCost^{(o)}$:
      $\hat{t}^{(o)} = \arg\min_{j \in \{0,1\}} \sum_{i=0}^{1} p_i^{(o)} c_{ij}$;
      $EstCost^{(o)} = \min_{j \in \{0,1\}} \sum_{i=0}^{1} p_i^{(o)} c_{ij}$;
    $sum\_cost = sum\_cost + EstCost^{(o)}$;
  end
  $avg\_cost = \dfrac{sum\_cost}{|O|}$
end

Algorithm 5: A general framework for dynamic label allocation

for all the objects and choose the one with the highest value, 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 good estimates of the labels for the objects that the worker has already worked on. This poses a disadvantage for the early-coming workers since they need to wait for a long time to get their quality correctly estimated. To alleviate the computational complexity and the latency in worker quality updates, we can divide the full set of objects into a number of subsets $N = \{N_1, N_2, \cdots, N_n\}$, where each N_i contains a relatively small number of objects.[8] We will start with the first subset N_1, and move to N_2 when the average estimated misclassification cost of N_1 is below the threshold τ_c, and so on. Note that as new labels arrive and the worker quality and object class probability estimates get updated, the estimated cost of previous batches might fail to meet the quality requirement. To overcome this problem, we allow labels to be allocated to previous batches when working on the current batch.

[8]The number of objects within each batch can be decided by the service provider. Smaller batches save computation time at the cost of suboptimal label resource allocation.

6 Simulation Experiments

To test the performance of the inference and label allocation strategies, we run a set of simulation experiments using synthetic data generated by the labeling model described in Section 3.2. Simulation experiments are a powerful tool for modeling complicated market environments and conducting analyses under various parameter values (e.g., Chiang and Mookerjee, 2004; Adomavicius et al., 2009; Ketter et al., 2012). We describe below the setting for the simulations.

The simulation setup is as follows: We have 1000 objects, evenly assigned to two classes. The easiness of each object $\beta^{(o)}$ is obtained by exponentiating a draw from a normal distribution $\mathcal{N}(0,1)$. There are 200 workers, whose quality parameters $\{\alpha_0^{(k)}\}$ and $\{\alpha_1^{(k)}\}$ are drawn from a normal distribution $\mathcal{N}(1,1)$.[9] The assigned labels $\{l_{(o)}^{(k)}\}$ are generated according to Equation (1) and (2). To test the performance of the algorithms under different cost settings, we employ two cost matrices: a symmetric cost matrix $\mathbf{c}^{(a)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and an asymmetric cost matrix $\mathbf{c}^{(b)} = \begin{pmatrix} 0 & 1 \\ 5 & 0 \end{pmatrix}$.[10] To smooth out variability between trials, the simulation is repeated 20 times and the results are averaged over all experimental runs.

6.1 Inference Algorithms in a Static System

We first look at a static system in which there is no adaptive decision making with respect to label allocation. Since there is no ex ante information, we generate equal number of labels for all objects. Based on the collected labels, the estimated class label (for MV and MP) or class probability estimates (for EM and GLAD) of each object, and the quality measure of each worker are obtained using different inference algorithms presented in Section 4. We evaluate the performance of these algorithms from two aspects: object actual misclassification cost and worker quality estimation accuracy. Since inference algorithms are not the main focus of this paper, for the sake of space, we present and discuss the results in Appendix D. The comparisons yield the following conclusions: (1) EM and GLAD outperform MV and MP by a large margin in both object actual misclassification cost and worker quality estimation accuracy; (2) GLAD results in a lower object actual misclassification cost than EM when the cost matrix is asymmetric; and (3) GLAD achieves a higher worker quality estimation accuracy than EM when the object assignment to each worker is not uniform with respect to easiness. Therefore, we use GLAD as the inference algorithm when testing the effectiveness of different label allocation strategies.

[9] The specific parameter values are chosen to produce similar level of label noise as in real-life scenarios. Our simulated datasets have an overall accuracy rate of 0.700, while the rates for the three real-world datasets *bluebird*, *rte*, and *temp* in Section 7 are 0.636, 0.729, and 0.734, respectively.

[10] We choose an asymmetric cost matrix $\mathbf{c}^{(b)} = \begin{pmatrix} 0 & 1 \\ 5 & 0 \end{pmatrix}$ to allow for a considerable, but not extreme, cost variation in different types of classification errors.

6.2 Label Allocation Strategies in a Dynamic System

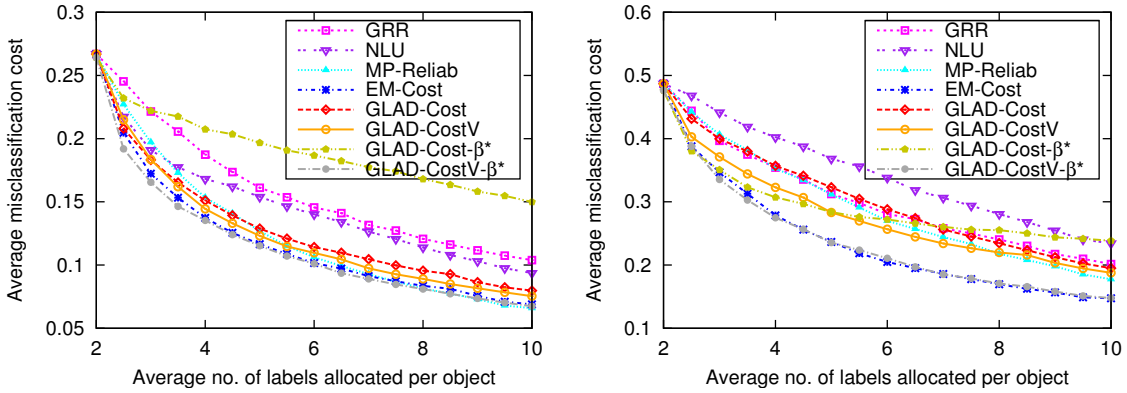
We now proceed to a dynamic system which allows the employer to allocate labels adaptively based on the data obtained so far. To mimic the dynamic process of the crowdsourcing marketplace, we assume that:[11] (1) assigning a label to an object requires 1 unit of time; (2) every 10 time units, a new worker comes to work on the available tasks; (3) each worker stops working once she contributes 50 labels.

As a baseline comparison, we implement a generalized round-robin (GRR) strategy which always assigns the next label to the object with the fewest number of labels, so that on average each object would receive equal number of labels. We also include the current state-of-the-art adaptive allocation strategy called new label uncertainty (NLU), presented by Ipeirotis et al. (2014), which assigns the next label to the object with the highest label uncertainty score, defined based on the posterior probability estimates of object class after obtaining a certain number of positive and negative labels. We test the performance of the four adaptive label allocation strategies (i.e., MP-Reliab, EM-Cost, GLAD-Cost, and GLAD-CostV) proposed in Section 5 against GRR and NLU. To ensure a fair comparison, we use GLAD as the inference algorithm for all the above-mentioned label allocation strategies.

Figure 2(a) reports the effectiveness of different allocation strategies under symmetric cost matrix $\mathbf{c}^{(a)}$, measured by the average actual misclassification cost of the objects. All the four strategies proposed in this paper show superior performance over GRR and NLU, with an improvement rate of $15\% - 35\%$ when the average number of labels allocated per object is 10. Among the four proposed strategies, EM-Cost achieves the best results overall; MP-Reliab performs poorly initially but catches up as more labels are collected; GLAD-CostV beats GLAD-Cost by a small margin. The simulation results under asymmetric cost matrix $\mathbf{c}^{(b)}$ are shown in Figure 2(b). NLU performs the worst, followed by GRR, MP-Reliab, and GLAD-Cost that produce similar results; GLAD-CostV has a clear advantage over GLAD-Cost; EM-Cost outperforms all the other strategies by a significant margin ($25\% - 45\%$).

The fact that EM-Cost outperforms GLAD-CostV is somewhat surprising, as we expect that allocating labels based on the expected variation of estimated cost will help to make better use of labeling resources. To explore the underlying mechanism driving these results, we check the performance of GLAD-Cost-β* and GLAD-CostV-β*, which are basically the same as GLAD-Cost and GLAD-CostV but assuming that the true easiness $\beta^{(o)}$ of each object (o) is known. It is clear from Figure 2(a) and Figure 2(b) that: (1) GLAD-Cost-β* performs very poorly, especially when the average number of labels allocated per object is large. This is probably due to the fact that GLAD-Cost-β* tends to allocate excessive labels to a small number of difficult objects that are more likely to have very high estimated misclassification costs. (2) GLAD-CostV-β* performs much better than GLAD-Cost-β*, confirming our intuition that cost variation

[11] Our results are robust to different specifications of worker arrival rate and lifetime.



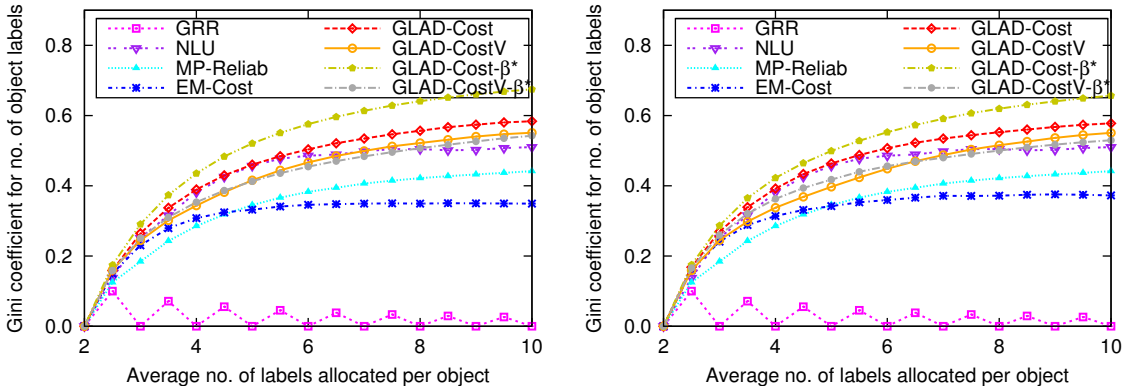
(a) symmetric cost matrix $\mathbf{c}^{(a)}$ (b) asymmetric cost matrix $\mathbf{c}^{(b)}$

Figure 2: Average actual misclassification cost as a function of the average number of labels acquired per object for different allocation algorithms in a dynamic system

is a good metric to use. However, GLAD-CostV-β* only brings marginal performance improvement over EM-Cost.

To see how labels are allocated among different objects, we introduce *Gini coefficient*, which measures the inequality of number of labels' distribution among objects. The Gini coefficient ranges from a minimum value of zero, when all objects receive equal number of labels, to a maximum value of one, when one object gets all the labels. The higher the Gini coefficient, the greater the degree of inequality in the distribution of labels across objects. We plot the Gini coefficients for different allocation strategies in Figure 3(a) and Figure 3(b). As expected, GRR has a near-zero Gini coefficient since it aims to equalize the number of labels assigned to each object. The Gini coefficient for GLAD-Cost-β* is very high, approaching 0.7 when the average number of labels allocated per object is 10. By taking into consideration the magnitude of cost variation, GLAD-CostV-β* is able to achieve a lower degree of inequality in label distribution. The Gini coefficients of GLAD-Cost and GLAD-CostV lie between the coefficients of GLAD-Cost-β* and GLAD-CostV-β* because of the imprecise estimates of object easiness. Notably, EM-Cost is associated with a moderate degree of inequality which also stabilizes as the average number of labels allocated per object increases.

Why does EM-Cost not suffer the same problem as GLAD-Cost and GLAD-Cost-β*? We turn to the basic assumption underlying EM algorithm, that is, workers' error rates do not change when labeling objects of varying degrees of easiness. The consequence is that EM is likely to produce overconfident (or extreme) class probability estimates for difficult objects (See Appendix E for an explanation). Therefore, the estimated misclassification cost of a difficult object under EM is likely to be lower than what is obtained using GLAD. As an object accrues many labels, its estimated misclassification cost under EM becomes very low, so the chance of this object being allocated with the next label is greatly reduced. The overconfident estimates act



(a) symmetric cost matrix $\mathbf{c}^{(a)}$ (b) asymmetric cost matrix $\mathbf{c}^{(b)}$

Figure 3: Gini coefficient for no. of labels' distribution among objects as a function of the average number of labels acquired per object for different allocation algorithms in a dynamic system

like a *penalty function* to prevent EM-Cost from over-investing labels in a few difficult objects and make it a pragmatic and effective strategy for allocating limited labels in a dynamic system.

7 Experiments on Real-World Crowdsourced Datasets

One drawback of using simulation is that the underlying label generation model is artificial, which is unlikely to hold in real-world settings. For example, the generation of labels may not follow a specific model; the errors workers make might be correlated. For further evaluation, we test the performance of the proposed approaches on three publicly available datasets obtained using Amazon Mechanical Turk. The first *bluebird* dataset is collected by Welinder et al. (2010), in which workers are asked whether the presented image contains Indigo Bunting or Blue GrosBeak. The second *rte* dataset and the third *temp* dataset are both natural language processing datasets collected by Snow et al. (2008): *rte* represents the recognizing textual entailment task, where workers are presented with two sentences and given a binary choice of whether the second hypothesis sentence can be inferred from the first; *temp* represents the event temporal annotation task, where workers are presented with a dialogue and a pair of verb events from the dialogue, and asked whether the event described by the first verb occurs before or after the second. Table 2 summarizes these three datasets.

All the three datasets above are collected in a static way, regardless of the labels acquired at each intermediate step. At the end of the data collection, all the objects receive the same number of labels. To simulate the dynamic label acquisition, we acquire labels in the following way:[12] At each step, we first pick a

[12]Such experiments based on real-world datasets possess two desirable properties: (1) The label generation process is real and does not assume any specific labeling model, which addresses the concerns of using artificial data. (2)

Dataset	No. of Objects (Positive/Negative)	No. of Workers	No. of Labels per Object	Mean/Median No. of Labels per Worker
bluebird	108 (60/48)	39	39	108/108
rte	800 (400/400)	164	10	49/20
temp	462 (259/203)	76	10	61/16

Table 2: The description of the three real-world datasets

worker from the set of workers who still have labels, with the probability of being picked proportional to the number of available labels per worker; then, we choose the next object to label based on the dynamic allocation strategies proposed in Section 5, with the constraint that the object must have been labeled by the chosen worker; next, we put the assigned label to the observed label set, and remove it from the label pool available for drawing. Same as before, we consider two cost settings $\mathbf{c}^{(a)}$ and $\mathbf{c}^{(b)}$ and average the results over 20 experimental runs.

7.1 Inference Algorithms

We first evaluate the performance of different inference algorithms in a scenario where all objects receive equal number of labels. The experimental results are shown in Figure F1, which confirm the superior performance of EM and GLAD. MP performs the worst on all three datasets and shows little or no improvement as more labels are allocated to each object. This is probably due to the fact that the regular graph assumption[13] of MP is violated in real-world settings where there exists both productive workers who tend to submit a large number of labels and unproductive workers who provide only a few labels. GLAD achieves similar results as EM when the cost matrix is symmetric, and slightly outperforms EM when the cost matrix is asymmetric. Therefore, we stick to GLAD as the inference algorithm for testing different allocation strategies.

7.2 Dynamic Label Allocation Strategies

Figure 4 reports the performance of different allocation strategies on real-world datasets, using GLAD as the inference algorithm. Clearly, EM-Cost, GLAD-Cost and GLAD-CostV perform consistently better than GRR, NLU, and MP-Reliab across all six combinations of datasets and cost settings. Different from what is observed on synthetic data, MP-Reliab does not even approach the performance of GRR on two of the datasets (i.e., *rte* and *temp*). This is because for both datasets, the number of objects labeled by each worker varies significantly: for *rte*, the number of labels contributed by each worker ranges from 20 to 800; and

All the different allocation strategies are using the same set of collected labels, allowing for a randomized controlled experiment that eliminates the influence of confounding factors across experimental runs. Chen et al. (2014) uses a similar approach in their paper for studying budget allocation.

[13] In a regular graph, each worker contributes equal number of labels, and each object receives equal number of labels.

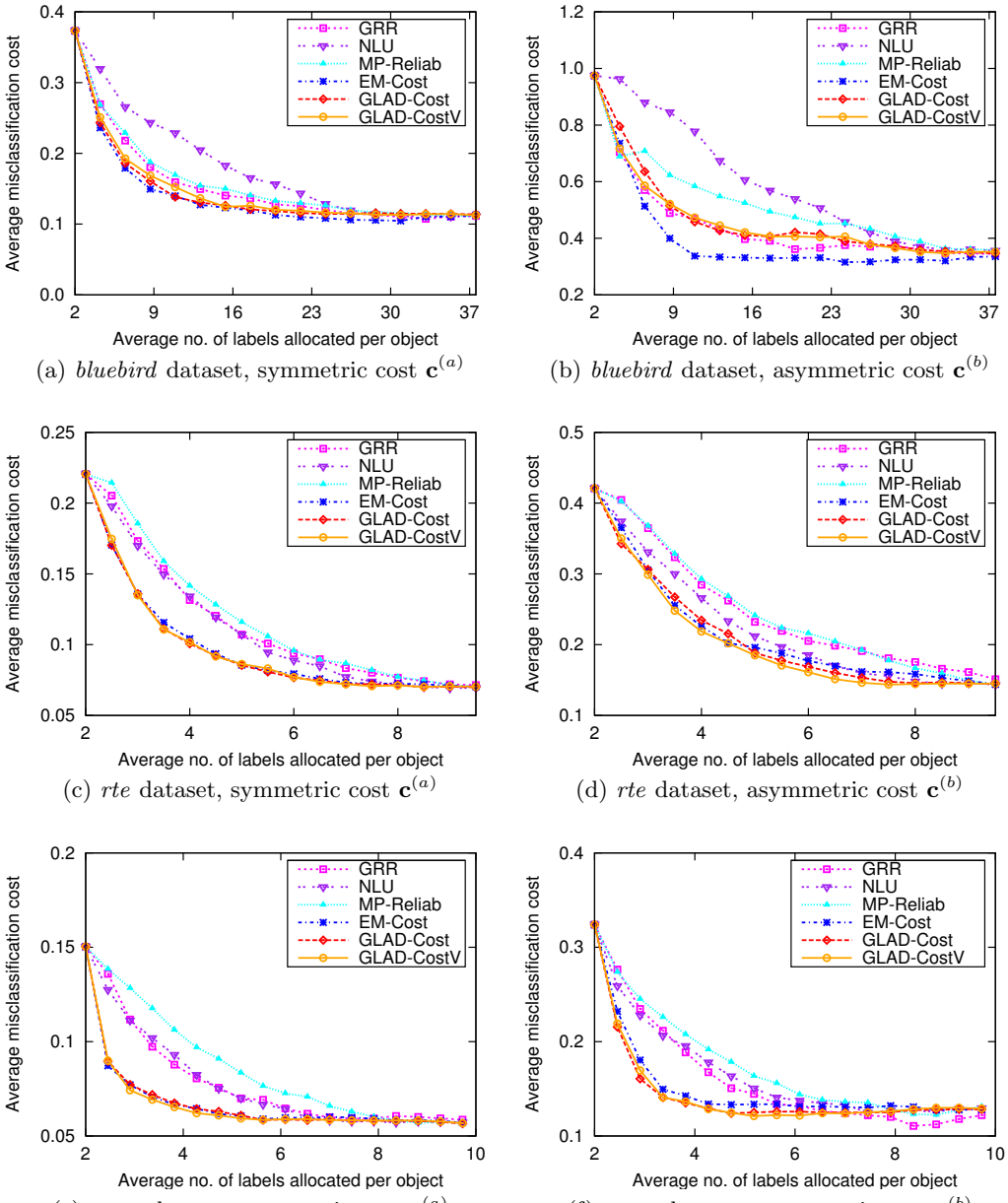
for *temp*, the range is between 10 and 462. In these scenarios, MP tends to weigh excessively the labels from productive workers and thus yield biased estimates for object classes. EM-Cost, GLAD-Cost and GLAD-CostV achieve similar performance in all cases except in Figure 4(b), where EM-Cost outperforms the other two strategies by a significant margin.

Note that the performance of the allocation strategies differs at first but converges as more labels are allocated to objects. This is because no matter what allocation strategy is employed, the labels are drawn from the same pool. At the beginning, each strategy has a considerable freedom of choice in allocating labels to objects; however, as more labels are allocated, some objects are running out of labels quickly and the next label has to been allocated to others; and at the end of the process, all the strategies get the same set of labels. Here, we report the performance improvement when the average number of labels allocated to each object is about one third of the total labels available to mitigate the interference of limited labels on evaluation. On *bluebird* dataset, EM-Cost outperforms the baseline GRR by 15% and NLU by 38% under $\mathbf{c}^{(a)}$, and the improvement rates are 23% and 50% under $\mathbf{c}^{(b)}$. On *rte* dataset, EM-Cost outperforms GRR by 25% and NLU by 23% under $\mathbf{c}^{(a)}$, and the rates are 21% and 15% under $\mathbf{c}^{(b)}$. On *temp* dataset, EM-Cost outperforms GRR by 27% and NLU by 31% under $\mathbf{c}^{(a)}$, and the rates are 29% and 28% under $\mathbf{c}^{(b)}$. We conclude that on these real-world datasets, our proposed allocation strategy EM-Cost can bring down the labeling cost by $15\% - 50\%$, which can be directly translated into huge economic savings when the number of objects to be classified is large.

8 Generating Reliable Worker Performance Metrics

In Appendix D.2, we leverage the advantage of simulated data to check the accuracy of worker quality estimation by calculating the Spearman coefficient between workers' true quality values and the estimated quality values using different inference algorithms.[14] The results show that both EM and GLAD achieve a fairly high level of accuracy and can be used as effective tools for evaluating worker quality. Unfortunately, neither the confusion matrix $\mathbf{e}^{(k)}$ returned by EM nor the quality vector $\hat{\boldsymbol{\alpha}}^{(k)}$ returned by GLAD is a scalar, and therefore cannot be directly used to rank worker performance. Here, we introduce two scalar metrics of worker performance: one is based on the estimated misclassification cost of a worker's label in single labeling (Section 8.2), and the other is based on the contributed value of a worker's label in multiple labeling (Section 8.3).

[14] The same procedure cannot be applied on real-world datasets since in practice workers' true quality values are always unknown.



(a) *bluebird* dataset, symmetric cost $\mathbf{c}^{(a)}$

(b) *bluebird* dataset, asymmetric cost $\mathbf{c}^{(b)}$

(c) *rte* dataset, symmetric cost $\mathbf{c}^{(a)}$

(d) *rte* dataset, asymmetric cost $\mathbf{c}^{(b)}$

(e) *temp* dataset, symmetric cost $\mathbf{c}^{(a)}$

(f) *temp* dataset, asymmetric cost $\mathbf{c}^{(b)}$

Figure 4: Average actual misclassification cost for different allocation algorithms on real-world datasets

8.1 EM-Equivalent Confusion Matrix for GLAD

The confusion matrix $\mathbf{e}^{(k)}$ produced by EM can fully reflect workers' errors in classifying objects of different classes. GLAD is more complicated in the sense that workers' errors not only depend on the quality vector $\hat{\boldsymbol{\alpha}}^{(k)}$ but also on the easiness of the object being classified. Therefore, the first step we take is to convert the quality vector into a measure that is able to capture the worker's overall classification errors independent of the specific object being labeled.

Based on Equation (1), the confusion matrix of a worker (k) on labeling a particular object (o) is $\hat{\boldsymbol{\xi}}^{(k,o)}$, where

$$\hat{\xi}_{ii}^{(k,o)} = \frac{1}{1+e^{-\hat{\alpha}_i^{(k)}\hat{\beta}^{(o)}}} \quad \text{and} \quad \hat{\xi}_{i,1-i}^{(k,o)} = 1 - \hat{\xi}_{ii}^{(k,o)}$$

The values in confusion matrix $\hat{\boldsymbol{\xi}}^{(k,o)}$ vary widely depending on the easiness of the object $\hat{\beta}^{(o)}$. If we simply take the average of $\hat{\boldsymbol{\xi}}^{(k,o)}$ over all the objects that worker (k) labels, we will overvalue a worker who labels disproportionately more easy objects and undervalue a worker who labels disproportionately more difficult objects. For a fair evaluation, we propose a measure $\bar{\boldsymbol{\xi}}^{(k)}$, where

$$\bar{\xi}_{ii}^{(k)} = \frac{1}{1+e^{-\hat{\alpha}_i^{(k)}\bar{\beta}}} \quad \text{and} \quad \bar{\xi}_{i,1-i}^{(k)} = 1 - \bar{\xi}_{ii}^{(k)}$$

Here, the $\bar{\beta}$ represents the average easiness of the objects, which is defined as $\bar{\beta} = \frac{1}{|O|}\sum_{(o)\in O}\hat{\beta}^{(o)}$. The element values of confusion matrix $\bar{\boldsymbol{\xi}}^{(k)}$ do not depend on the specific objects assigned to worker (k) and thus can objectively evaluate worker performance. For ease of presentation, below we use $\mathbf{e}^{(k)}$ to denote both $\mathbf{e}^{(k)}$ produced by EM and $\bar{\boldsymbol{\xi}}^{(k)}$ produced by GLAD.

8.2 Estimated Cost of a Worker in Single-Label Case

A straightforward method for evaluating worker performance is to calculate the accuracy rate (i.e., how often the worker submits a correct label) for each worker based on $\mathbf{e}^{(k)}$. However, this approach may mistakenly reject workers whose labels are wrong but informative. Consider the following example:

Example 5 *Two workers are working on the task of classifying web sites into two groups:* porn *and* notporn*. Worker A is always incorrect, labeling all* porn *web sites as* notporn *and vice versa. Worker B is lazy and classifies all web sites as* porn*. A simple analysis indicates that the accuracy rate of worker A is 0%, while the accuracy rate of worker B is only 50%.*[15] *However, it is not difficult to see that worker A's errors are easily reversible, while worker B's errors are irreversible.*

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

Input: Confusion matrix $\mathbf{e}^{(k)}$, misclassification cost matrix $\mathbf{c}$, estimated class prior vector $\hat{\boldsymbol{\pi}}$
Output: Estimated cost $EstCost^{(k)}$ of each worker (k)

foreach *worker* (k) **do**
  $EstCost^{(k)} = 0$;
  foreach *hard* label l **do**
    Estimate the prior probability that worker (k) assigns label l: $\hat{\pi}_l^{(k)} = \sum_{i=0}^{1} \hat{\pi}_i e_{il}^{(k)}$;
    Compute the posterior *soft* label vector corresponding to *hard* label l:
      $\mathbf{soft}^{(k)}(l) = (\frac{\hat{\pi}_0 e_{0l}^{(k)}}{\hat{\pi}_l^{(k)}}, \frac{\hat{\pi}_1 e_{1l}^{(k)}}{\hat{\pi}_l^{(k)}})$;
    Using Proposition 1, compute $EstCost(\mathbf{soft}^{(k)}(l))$ for the *soft* label;
    $EstCost^{(k)}$ += $EstCost(\mathbf{soft}^{(k)}(l)) \cdot \hat{\pi}_l^{(k)}$;
  end
end
return $EstCost^{(k)}$ *for each worker* (k)

Algorithm 6: Calculating the Estimated Cost of each Worker

Another drawback of using accuracy rate is that all types of classification errors are treated indiscriminately. However, in reality, some errors can be more costly than others. For the classification task in Example 5, labeling a *porn* website as *notporn* can lead to serious consequences while labeling a *notporn* website as *porn* is less harmful.

Naturally, a questions arises: Given the estimates of confusion matrix $\mathbf{e}^{(k)}$ for each worker (k), how can we generate a reliable worker performance metric that can separate correctable errors from uncorrectable errors workers make, and weight different types of errors based on their cost magnitudes?

Each worker assigns a *hard* label to each object. Using the confusion matrix of this worker, we can transform this assigned label into a *soft* label (i.e., posterior estimate), which is the best possible probability estimate we have for the true class of the object. If the worker (k) assigns l as the label to an object, we can transform this *hard* assigned label into a posterior *soft* label vector $(\hat{\pi}_0 e_{0l}^{(k)}, \hat{\pi}_1 e_{1l}^{(k)})$, where $\hat{\pi}_0$ and $\hat{\pi}_1$ are the estimated class priors. Of course, the quantities above need to be normalized by dividing them with $\hat{\pi}_l^{(k)} = \sum_{i=0}^{1} \hat{\pi}_i e_{il}^{(k)}$, which denotes the estimated prior probability that worker (k) assigns a label l. The misclassification cost of this *soft* label can be estimated based on Proposition 1.

Knowing how to compute the estimated prior probability that worker (k) assigns each *hard* label, and the estimated cost of the posterior *soft* label vector corresponding to the *hard* label, we can easily calculate the estimated cost of worker (k). Algorithm 6 illustrates the process.

Example 6 *Consider the estimated costs of workers A and worker B from the previous example. 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 are* $(0.0, 1.0)$ *and* $(1.0, 0.0)$ *when the hard labels provided by A are 0 and 1. For worker B, the cost is 0.5 (the maximum possible) as the soft label generated by B is always* $(0.5, 0.5)$.

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) tend to have low estimated cost. On the contrary, workers that generate posterior labels with probabilities widely spread across classes (i.e., uncertain posterior labels) tend to have high misclassification costs. This performance metric based on estimated misclassification cost resolves quite a few issues of prior approaches that rely on agreement, which generate a significant number of rejections for workers whose labels are wrong but informative and workers whose errors do not incur a high cost. However, it only works in the single label case where each object only receives one label. We next discuss how to evaluate the performance of a worker in a multiple-label setting.

8.3 Contributed Value of a Worker in Multiple-Label Case

As mentioned in Section 3.1, the ultimate objective of the employer is to get all objects labeled with average misclassification cost not exceeding a threshold τ_c. For ease of exposition, we define a worker as *qualified* worker if her estimated cost is below τ_c; otherwise, the worker is considered an *unqualified* worker. Since the data quality requirement is usually high, many workers in crowdsourcing markets fall into the category of unqualified workers. In fact, there might be cases where *no* worker satisfies the desired quality. Simply considering these workers as having no value and disregarding their labels is short-sighted and renders the problem intractable. Although each individual worker does not necessarily submit high-quality labels, a group of them as a whole may be able to achieve the requirements. A substantial number of papers in the literature (e.g., Sheng et al., 2008; Snow et al., 2008; Welinder et al., 2010; Raykar et al., 2010; Ipeirotis et al., 2010; Bachrach et al., 2012) have shown that multiple low-quality workers can work in tandem to generate results of high quality. The focus of this section is to derive the value of such *unqualified* workers, according to the level of redundancy required to reach the required quality standards.

Example 7 *Suppose an employer has a binary classification problem with equal class priors, misclassification cost set to* 1*, and a quality requirement that the average misclassification cost is below* 0.1*. The employer gains* \$1 *value from each classified object meeting the required quality level. 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 quality objective? Figure 5 shows the relationship between the number of workers and the integrated estimated cost with the value of* q *ranging from* 0.60 *to* 0.90 *at an interval of* 0.05*. The black dash line indicates the required cost level. We can see that:*

1. *A worker with* $q = 0.90$ *is a* qualified *worker, and is worth* \$1 *to the service provider.*
2. *A worker with* $q = 0.80$ *is* unqualified*. However, a set of* 3 *workers with* $q = 0.80$ *generate labeling of required quality. Therefore a worker with* $q = 0.80$ *is worth* \$0.33*.*

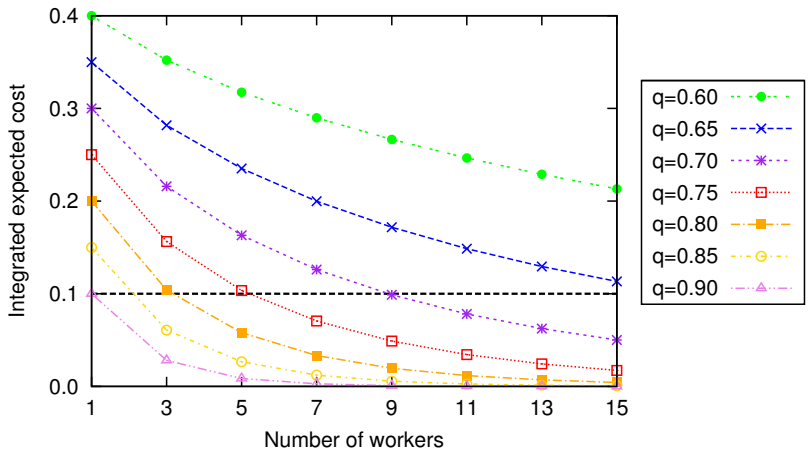


Figure 5: The relationship between the number of workers and integrated estimated cost

3. *A worker with* $q = 0.70$ *is* unqualified. *We need 9 workers with* $q = 0.70$ *to reach the required quality, therefore a worker with* $q = 0.70$ *is worth* \$0.11.

Therefore, the contributed *value* of a worker is inversely proportional the number of workers with the same confusion matrix required to achieve the acceptable quality level. Next, we show the process for estimating the value of a worker with an arbitrary confusion matrix $\mathbf{e}$.

Definition 8 *The* value $v(\mathbf{e})$ *of a worker with confusion matrix* $\mathbf{e}$ *is:* $v(\mathbf{e}) = \dfrac{V}{d(\mathbf{e})}$, *where* $d(\mathbf{e})$ *is the number of workers with confusion matrix* $\mathbf{e}$ *required to reach the target average misclassification cost* τ_c, *and* V *is the value that the employer can gain from a unit of object with acceptable cost level. For qualified workers* $d(\mathbf{e}) = 1$, *while for unqualified workers* $d(\mathbf{e}) > 1$.

Now 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 $\mathbf{e}$ required to generate labeling of acceptable quality. 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, because of the exchangeability of the labels, can be represented as a count of all class labels $\mathbf{n} = \{n_0, m - n_0\}$. When the true class label is i (which occurs with probability $\hat{\pi}_i$), this label assignment happens with probability $f(n_0; m, e_{i0}) = \binom{m}{n_0} (e_{i0})^{n_0} (1 - e_{i0})^{m - n_0}$, which is the probability mass function (pmf) of the binomial distribution with parameters m (number of trials) and e_{i0} (success probability in each trial). Integrating this over both classes, we get the overall probability of seeing $\mathbf{n}$ is:

$$p(\mathbf{n}) = \sum_{i=0}^{1} \hat{\pi}_i f(n_0; m, e_{i0}) = \binom{m}{n_0} \hat{\pi}_i (e_{i0})^{n_0} (1 - e_{i0})^{m - n_0} \tag{6}$$

For each label assignment $\mathbf{n} = \{n_0, m - n_0\}$, the *soft* label before normalization is proportional to:

$$\left(\hat{\pi}_0 (e_{00})^{n_0} (1 - e_{00})^{m-n_0}, \hat{\pi}_1 (e_{10})^{n_0} (1 - e_{10})^{m-n_0}\right) \tag{7}$$

The estimated misclassification cost associated with the label assignment $\mathbf{n}$ is then calculated using Proposition 1. By repeating the process across all possible label assignments and weighting the cost of each one by its occurrence probability, we get the average misclassification cost of using m workers with confusion matrix $\mathbf{e}$. Knowing how to compute the integrated estimated cost, the value derivation becomes quite easy. Given a worker with specific confusion matrix $\mathbf{e}$, we simply find the minimum number of workers $d(\mathbf{e})$ needed to achieve the required cost level.

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 $d(\mathbf{e})$. In addition, the $d(\mathbf{e})$ generated above is likely to be an overestimate as we force each label assignment to have equal number of labels. As illustrated in the Section 6 and 7, selective label acquisition can potentially reduce the amount of labels required to achieve the target quality level. Therefore, we resort to a Monte Carlo approach for estimating $d(\mathbf{e})$, where labels are drawn incrementally and prioritized to objects with high estimated misclassification costs, allowing some types of label assignments to have more labels than others. Algorithm 7 illustrates the overall process.[16] Note that the worker values $v(\mathbf{e})$ can be computed beforehand and stored in a two-dimensional matrix.[17] The number of elements in the matrix determines the degree of accuracy. For example, if we round e_{00} and e_{11} to one decimal place, the number of elements is $11 \times 11 = 121$; if we round e_{00} and e_{11} to two decimal places, the number of elements is $101 \times 101 = 10201$.

To see how the contributed value metric differs from the naive accuracy rate metric, we present workers' performance based on accuracy rate (x-axis) and contributed value (y-axis) in Figure 6, estimated from the three real-world datasets in Section 7. Each blue dot represents a worker. For easier comparison, we normalize both measures by the sum of values of all workers so that the two measures are on the same scale. We also draw a red diagonal line with x and y having the same value. As shown in the figure, there is a notable discrepancy between the accuracy rate measure and the contributed value measure. Overall, using accuracy rate as a performance measure tends to overestimate the contribution of low-quality workers and underestimate the contribution of high-quality workers.

[16] We use EM-Cost because of its outstanding performance on both simulated and real-world datasets. To save computation cost without sacrificing too much accuracy, we set $D = 30$ and $N = 1000$ in the actual implementation.

[17] Since e_{00} and e_{11} fully determine $\mathbf{e}$, we can use them as row- and column-index and put $v(\mathbf{e})$ into the corresponding matrix element.

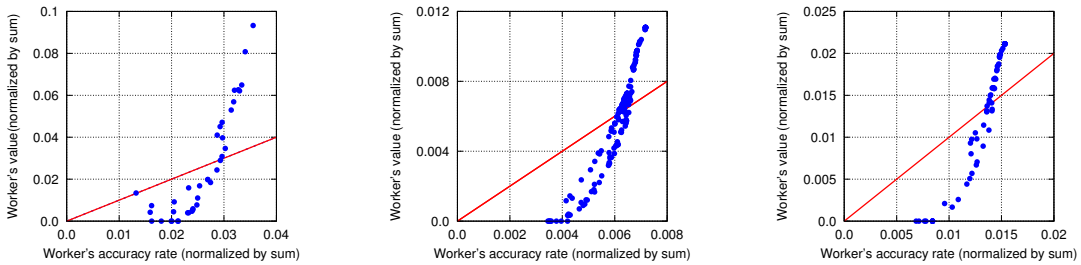
Input: Confusion matrix $\mathbf{e}$, misclassification cost matrix $\mathbf{c}$, estimated class prior vector $\hat{\boldsymbol{\pi}}$, unit price for qualified objects V, sample size N, maximum number of workers D
Output: Value $v(\mathbf{e})$

for $x = 1$ *to* N **do**
  Generate object x with the true class drawn from prior vector $\hat{\boldsymbol{\pi}}$;
  Using Proposition 1, compute $EstCost(x)$ based on prior probability vector $\hat{\boldsymbol{\pi}}$;
end
$cnt = 0$;
while $cnt \leq D \cdot N$ **do**
  Pick the object y with the highest estimated cost (i.e., $EstCost(y) \geq EstCost(x), \forall x$);
  Draw one label for object y, following confusion matrix $\mathbf{e}$;
  $cnt = cnt + 1$;
  Using Equation (7), compute the posterior probability vector $\mathbf{p}(y)$ for object y;
  Using Proposition 1, compute $EstCost(y)$ for the posterior probability vector $\mathbf{p}(y)$;
  $sum\_cost = 0$;
  for $x = 1$ *to* N **do**
    $sum\_cost = sum\_cost + EstCost(x)$;
  end
  $avg\_cost = \dfrac{sum\_cost}{N}$;
  if $avg\_cost \leq \tau_c$ **then**
    break;
  end
end
if $cnt \leq D \cdot N$ **then**
  $d(\mathbf{e}) = \dfrac{cnt}{N}$; $v(\mathbf{e}) = \dfrac{V}{d(\mathbf{e})}$;
else
  $v(\mathbf{e}) = 0$;
end

Algorithm 7: Estimating the value $v(\mathbf{e})$ of a worker with confusion matrix $\mathbf{e}$

As a leading player in micro-crowdsourcing markets, AMT has implemented operations such as *GrantBonus*[18] and *BlockWorker*[19] to supplement its piece-rate compensation system. We believe that the contributed value measure developed in this paper, which represents the monetary value that the employer can derive from each label of a worker, can be used as a basis for employers to grant bonuses to good workers and block low-value workers from further participation. For example, a simple way to utilize this measure is to offer a base payment for all workers and block those workers whose contributed value is below the base payment. A more advanced usage is to pay a bonus on top of the piece-rate based on the performance of workers.[20] Figure F2 illustrates how the bonus payment interface looks like to workers. Workers are shown their current bonus levels throughout their participation and they can move the cursor around to know the bonus amounts associated with other performance levels. The granularity of the performance measure can be adjusted to suit workers' comprehension level.

[18] http://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_GrantBonusOperation.html
[19] http://docs.aws.amazon.com/AWSMechTurk/latest/AWSMturkAPI/ApiReference_BlockWorkerOperation.html
[20] See Ho et al. (2015) for an example of using bonus payments to induce high-quality work.



(a) *bluebird* dataset (b) *rte* dataset (c) *temp* dataset

Figure 6: Scatter plots of worker's accuracy rate (normalized by sum) versus worker's contributed value (normalized by sum) on real-world datasets

To get a preliminary idea of how a quality-sensitive payment scheme works, we conducted the following experiment, in a live production system, where workers were asked to take a look at online discussion forums, and decide whether the question discussed in the forum can be used as an assessment question in an exam.[21] The workers were automatically hired using an API from the oDesk labor platform. In the control condition, all workers were paid the same but when a worker's performance dropped below a particular threshold (which is set to 80%), her contract would be terminated. In the experimental treatment, workers got paid under quality-based pricing (QBP) and received compensation proportional to the value they were adding to the system.[22] A total of 120 workers participated in the experiment, split in equal parts among the treatment and control conditions.

The results indicated a significant improvement of QBP not only in terms of cost reduction but also in terms of worker retention. The cost for completing the tasks at the same level of quality was 50% to 70% lower in the QBP condition compared to the control. Furthermore, the lifetime of the workers in the QBP condition was 1.5 to 3 times longer when measured by the number of completed tasks; in other words, the churn rate in the control condition was significantly higher. The high churn is undesirable both from a human management perspective and from a data gathering and statistics perspective: For many workers in the control condition, the collected data were insufficient to allow for accurate inferences about their quality, which in turn led to the need to collect more data, to compensate for this uncertainty. We believe that the results of this experiment highlight the potential for quality-based pricing, not only as a cost-saving scheme, but also as a means to improve worker retention.

Although the results are encouraging, we acknowledge that determining the optimal amount of bonus to offer to each worker is a complex problem, which may depend on a number of factors, including the reservation wages and risk preferences of workers, the type of micro-tasks (skill- or effort-based), the demand-

[21] For more details about the system, please see Christoforaki and Ipeirotis (2014).

[22] For example, a worker who was 80% accurate, was paid one third of the price offered to a worker who was 90% accurate, as we need three workers with 80% accuracy to *simulate* a worker with 90% accuracy.

supply balance of crowdsourced workers, the motivation component of workers (intrinsic or extrinsic), etc. Nevertheless, our proposed contributed value metric can tell the employer approximately how much monetary value she can derive from the labels provided by workers and make a step further towards the development of more fair and efficient compensation systems.

9 Conclusions and Discussion

Crowd labeling has rapidly become a commonly used tool for companies and researchers to acquire a huge number of cheap labels. However, such non-expert labels are often noisy and unreliable, and employers need to rely on redundancy to achieve a desired level of quality, which significantly increases the total labeling expense. Therefore, devising cost-effective label acquisition strategies and establishing reliable worker performance metrics are of substantial interest to decision makers.

The contribution of this paper is twofold: First, we formulate a dynamic decision system in which label allocation and inference occur simultaneously, and propose several adaptive label allocation strategies that prioritize labels on objects that are more likely to yield higher rewards for employers. We demonstrate the superior performance of the proposed strategies over alternative approaches via extensive experiments on both simulated and real-world datasets. Second, we introduce two novel metrics that can be used to objectively rank the performance of crowdsourced workers, both allowing employers to separate workers' correctable errors from uncorrectable errors and incorporate unequal costs of different types of classification errors. In particular, the contributed value metric directly measures worker's individual contribution in quality assurance through redundancy and provides a basis for employers to develop more fair and efficient compensation schemes. As illustrated further in Appendix A, our work may serve as a fundamental quality control block for a variety of tasks, ensuring that the outcome of crowdsourced production reaches the quality levels desired by employers.

9.1 Practical Implications

Despite the wide adoption of micro-crowdsourcing by various companies, quality assurance at minimal cost remains an issue yet to be explored. Many of today's firms still operate on a static system in which a fixed number of labels are collected for each object first, and some aggregation method is then employed to infer the true classes of objects. However, the real-world crowd labeling system is inherently dynamic, which provides an opportunity for companies to allocate labels adaptively and efficiently so that the target data quality can be achieved at considerably less expense. As illustrated in previous experiments, compared with the non-adaptive scheme, our proposed label allocation strategies can reduce the labeling expense by $15\% - 50\%$.

For big companies (e.g., Facebook, Twitter) that require tons of human labels on an everyday basis, the implementation of such adaptive schemes can help to save millions of dollars in data acquisition costs.

Crowdsourcing also lowers the barrier-to-entry for workers and provides a great way to help unemployed and underemployed people. Since there is no interview stage and workers can join the workforce at will, employers often face a pool of heterogeneous workers. Our proposed worker performance metrics can reliably assess the performance of each worker and distinguish informative workers from those of little use. The approach of evaluating workers based on their contributed value towards a desired level of data quality can facilitate the implementation of a bonus-based compensation scheme to motivate crowdsourced workers to submit more high-quality work and foster the creation of a healthy, well-operating crowdsourcing marketplace.

9.2 Limitations and Future Work

This study has several limitations and opens up opportunities for further research. First, in our study, we assume that worker quality does not change over time. However, for many types of tasks in practice, there might be either learning effects or tiredness effects, which may lead to possible fluctuations in the exhibited quality of workers. To account for this, we can apply a particle filtering method to track the changes in worker quality (Crisan and Doucet, 2002; Donmez et al., 2010) and choose the size of window for aggregation appropriately (Aperjis and Johari, 2010).

Second, in reality, sometimes the employer has access to the past performance of workers on the same or similar tasks. Kokkodis and Ipeirotis (2015) show that worker reputation is transferable across categories and predictive of future performance. Knowledge of prior, intra- or inter-category reputation of workers could potentially improve the estimation accuracy of worker quality, especially when the worker only submits a very few labels. As demonstrated in Section 4.3 and 4.4, EM and GLAD algorithms can work either independently or in tandem with the existence of a reputation system.

Third, the focus of this paper is to guarantee a certain level of labeling quality at as low cost as possible. With the aid of advanced supervised learning techniques, companies can use a small set of labeled objects to build classification models to make predictions on the set of unlabeled objects. In this case, the optimization problem becomes how to achieve a desirable level of model predictive performance at minimum label acquisition cost. To solve this problem, we can adjust the dynamic label allocation strategies by taking into consideration the prediction uncertainty of objects and prioritize labels to objects that are likely to induce greater improvement in both data and model quality.

Fourth, we discuss the potential of our worker value metric in guiding the design of more effective compensation schemes and present some preliminary real-world experimental results to illustrate the benefits of quality-based pricing in reducing labeling cost and improving worker retention. However, since there are

many factors at play, coming up with a specific compensation contract that can be used immediately by all employers to achieve optimal profits is extraordinarily difficult. For example, Ho et al. (2015) show that bonus payments only work when tasks are effort-responsive. A comprehensive examination of how workers respond to different incentives requires extensive experimentation across a wide range of task settings and is thus beyond the scope of this paper.

Despite these limitations, we believe that our current work provides a solid foundation on which future work can build. The proposed dynamic label allocation strategies substantially reduce the labeling expenses incurred by employers, and thus contribute to better and efficient utilization of crowd intelligence. Our value-based worker performance metric gives a fairly reasonable estimate of the contribution of individual workers in monetary terms and provides reference for employers to offer performance-contingent bonuses to motivate crowdsourced workers. Furthermore, our work can be used immediately by interested parties, allowing easier management of crowdsourced workers, and therefore the development of more interesting applications, enabled by micro-crowdsourcing.

References

Adomavicius, Gediminas, Alok Gupta, Dmitry Zhdanov. 2009. Designing intelligent software agents for auctions with limited information feedback. *Information Systems Research* **20**(4) 507–526.

Aperjis, Christina, Ramesh Johari. 2010. Optimal windows for aggregating ratings in electronic marketplaces. *Management Science* **56**(5) 864–880.

Archak, Nikolay, Anindya Ghose, Panagiotis G Ipeirotis. 2011. Deriving the pricing power of product features by mining consumer reviews. *Management Science* **57**(8) 1485–1509.

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/`.

Chen, Xi, Qihang Lin, Dengyong Zhou. 2014. Statistical decision making for optimal budget allocation in

crowd labeling. *Available at SSRN 2408163* .

Chiang, I Robert, Vijay S Mookerjee. 2004. A fault threshold policy to manage software development projects. *Information Systems Research* **15**(1) 3–21.

Christoforaki, Maria, Panagiotis Ipeirotis. 2014. Step: A scalable testing and evaluation platform. *Second AAAI Conference on Human Computation and Crowdsourcing*.

Cohn, David, Les Atlas, Richard Ladner. 1994. Improving generalization with active learning. *Machine learning* **15**(2) 201–221.

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. 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.

Goes, Paulo B. 2014. Editor's comments: design science research in top information systems journals. *MIS Quarterly* **38**(1) iii–viii.

Gregor, Shirley, Alan R Hevner. 2013. Positioning and presenting design science research for maximum impact. *MIS quarterly* **37**(2) 337–356.

Hevner, Alan R., Salvatore T. March, Jinsoo Park, Sudha Ram. 2004. Design science in information systems research. *MIS quarterly* **28**(1) 75–105.

Ho, Chien-Ju, Aleksandrs Slivkins, Siddharth Suri, Jennifer Wortman Vaughan. 2015. Incentivizing high quality crowdwork. *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 419–429.

Ipeirotis, Panagiotis G. 2010. Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students* **17**(2) 16–21.

Ipeirotis, Panagiotis G, Foster Provost, Victor S Sheng, Jing Wang. 2014. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery* **28**(2) 402–441.

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.

Karger, David R, Sewoong Oh, Devavrat Shah. 2011. Iterative learning for reliable crowdsourcing systems. *Advances in neural information processing systems*. 1953–1961.

Ketter, Wolfgang, John Collins, Maria Gini, Alok Gupta, Paul Schrater. 2012. Real-time tactical and strategic sales management for intelligent agents guided by economic regimes. *Information Systems Research* **23**(4) 1263–1283.

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.

Kokkodis, Marios, Panagiotis G Ipeirotis. 2015. Reputation transferability in online labor markets. *Management Science* .

Kuechler, William, Vijay Vaishnavi. 2012. A framework for theory development in design science research: multiple perspectives. *Journal of the Association for Information systems* **13**(6) 395–423.

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.

Lewis, David D, William A Gale. 1994. A sequential algorithm for training text classifiers. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag New York, Inc., 3–12.

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.

Lizotte, Daniel J, Omid Madani, Russell Greiner. 2002. Budgeted learning of nailve-bayes classifiers. *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 378–385.

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

March, Salvatore T, Veda C Storey. 2008. Design science in the information systems discipline: an introduction to the special issue on design science research. *Management Information Systems Quarterly* **32**(4) 6.

Moore, James C, Andrew B Whinston. 1986. A model of decision-making with sequential information-acquisition (part 1). *Decision Support Systems* **2**(4) 285–307.

Moore, James C, Andrew B Whinston. 1987. A model of decision-making with sequential information-acquisition (part 2). *Decision Support Systems* **3**(1) 47–72.

Moreno, Antonio, Christian Terwiesch. 2014. Doing business with strangers: Reputation in online service marketplaces. *Information Systems Research* **25**(4) 865–886.

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* **11** 1297–1322.

Roy, Nicholas, Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. *Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., 441–448.

Saar-Tsechansky, Maytal, Prem Melville, Foster Provost. 2009. Active feature-value acquisition. *Management Science* **55**(4) 664–684.

Saar-Tsechansky, Maytal, Foster Provost. 2004. Active sampling for class probability estimation and ranking. *Machine learning* **54**(2) 153–178.

Saar-Tsechansky, Maytal, Foster Provost. 2007. Decision-centric active learning of binary-outcome models. *Information Systems Research* **18**(1) 4–22.

Schilling, Edward G. 1982. *Acceptance Sampling in Quality Control*. CRC Press.

Sheng, V. S., F. Provost, P. G. Ipeirotis. 2008. Get another label? Improving data quality and data mining using multiple, noisy labelers. *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2008)*. 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.

Wais, Paul, Shivaram Lingamneni, Duncan Cook, Jason Fennell, Benjamin Goldenberg, Daniel Lubarov, David Marin, Hari Simons. 2010. Towards building a high-quality workforce with Mechanical Turk. *Proceedings of Computational Social Science and the Wisdom of Crowds (NIPS)*. 1–5.

Wang, Jing, Anindya Ghose, Panos Ipeirotis. 2012. Bonus, disclosure, and choice: What motivates the creation of high-quality paid reviews? .

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

Welinder, Peter, Pietro Perona. 2010. Online crowdsourcing: rating annotators and obtaining cost-effective labels. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 25–32.

Wetherill, G. B., W. K. 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.

Zheng, Zhiqiang, Balaji Padmanabhan. 2006. Selectively acquiring customer information: A new data acquisition problem and an active learning-based solution. *Management Science* **52**(5) 697–712.

Appendix A Importance of Quality Control for Binary Choice Questions

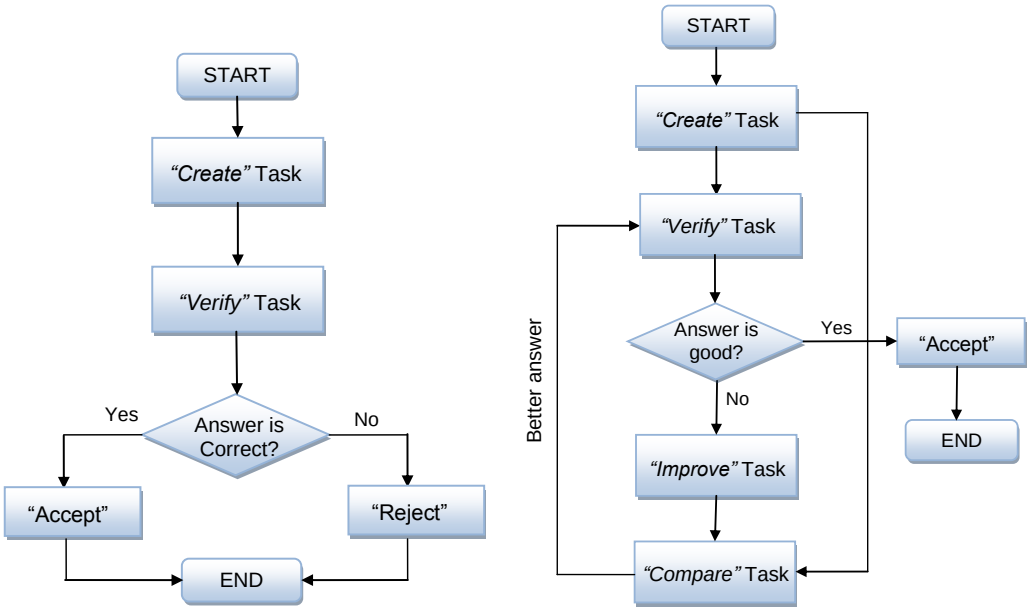
Our scheme can be directly applied to binary choice questions, which already captures a large number of tasks that are crowdsourced today (e.g., sentiment judgement, spam detection). We would like to stress, though, that quality control mechanisms for binary choice questions are at the heart of many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

- **Open-ended questions with correct or incorrect answers**: Consider the task of collecting 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 difficult or infeasible to enumerate all the correct answers, therefore it is not possible to control the quality of the task using the quality control mechanism for binary 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, one can break the task into two subtasks: a *"Create"* task in which one or more workers submit free-form answers, and a *"Verify"* task in which another set of workers vet the submitted answers, and classify them as either correct or incorrect. Figure A1(a) illustrates the structure: "Verify" task controls the quality of "Create" task; the quality of "Verify" task is then controlled using a quality control mechanism for binary choice questions, similar to the one presented in this paper.
- **Varying degrees of correctness**: There are some tasks whose free-form answers are not right or wrong but have varying 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 might be inefficient: a rejected answer would be completely discarded, although it is often possible to leverage 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 that workers are able to create image descriptions of excellent quality, even though no single worker puts any significant effort. Figure A1(b) illustrates the iterative process. There are four subtasks: a *"Create"* task in which free-form answers are submitted, an *"Improve"* task in which workers are asked to improve an existing answer, a *"Compare"* task in which workers are required to compare two answers and select the better one, and a *"Verify"* task in which workers decide whether the quality of the answers[23] is satisfactory. In this case, "Compare" task and "Verify" task are binary choice tasks, and one can use the mechanisms presented in this paper to control the quality

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

of the submitted answers (and of the participating workers). In turn, the quality of "Create" task and "Improve" task is controlled by "Verify" and "Compare" tasks, as one can measure the probability that a worker submits an answer of high quality, or the probability that a worker is able to improve an existing answer.

- **Complex tasks using workflows**: Initial applications of paid micro-crowdsourcing focused primarily 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 micro-tasks, using workflows. For example, Bernstein et al. (2010) introduce 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, and verify the correctness of the fix. Again, this task ends up having quality control through a set of binary choice tasks (verification of the fix, verification that something needs fixing). In other cases, Kittur et al. (2011) describe 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 into 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.



(a) Correct or incorrect answers

(b) Varying degrees of correctness

Figure A1: Workflows for two types of tasks

Appendix B Full GLAD Derivation

In our model, the set of labels $L = \{l_{(o)}^{(k)}\}$ are known, while the quality of each worker (k), the easiness of each object (o) and the true class of each object (o) are unknown and have to be estimated from the set of given labels.

Following Whitehill et al. (2009), we use expectation-maximization (EM) approach to obtain the maximum likelihood estimates of the $\boldsymbol{\alpha}^{(k)}$, $\beta^{(o)}$, and $t^{(o)}$ for each worker (k) and each object (o).

E-step: The posterior probability of $t^{(o)}$ given $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\beta^{(o)}\}$ is characterized by:

$$
\begin{aligned}
p\big(t^{(o)}|L,\{\boldsymbol{\alpha}^{(k)}\},\{\beta^{(o)}\}\big) &= p\big(t^{(o)}|L^{(o)},\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\},\beta^{(o)}\big) \\
&\propto p\big(t^{(o)}|\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\},\beta^{(o)}\big)p\big(L^{(o)}|t^{(o)},\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\},\beta^{(o)}\big) \\
&\text{since } l_{(o)}^{(k)}\text{'s are cond. indep. given } t^{(o)},\{\boldsymbol{\alpha}^{(k)}\} \text{ and } \beta^{(o)} \\
&\propto p(t^{(o)}) \prod_{(k)\in K^{(o)}} p\big(l_{(o)}^{(k)}|t^{(o)},\alpha_{t^{(o)}}^{(k)},\beta^{(o)}\big) \\
&\propto p(t^{(o)}) \prod_{(k)\in K^{(o)}} \Big(\frac{1}{1+e^{-\alpha_{t^{(o)}}^{(k)}\beta^{(o)}}}\Big)^{I\left(l_{(o)}^{(k)}=t^{(o)}\right)} \Big(\frac{1}{1+e^{\alpha_{t^{(o)}}^{(k)}\beta^{(o)}}}\Big)^{I\left(l_{(o)}^{(k)}=1-t^{(o)}\right)}
\end{aligned}
\tag{8}
$$

Following equation (8), we can calculate the posterior probability of $t^{(o)}$ using the prior probability of $t^{(o)}$, the values of $\{\boldsymbol{\alpha}^{(k)}|(k)\in K^{(o)}\}$ and the value of $\beta^{(o)}$ estimated from the previous M-step.

M-step: We maximize the auxiliary function Q, which is defined as the expectation of the joint log-likelihood of the observed and hidden variables $(L,\{t^{(o)}\})$ given the parameters $(\{\boldsymbol{\alpha}^{(k)}\},\{\beta^{(o)}\})$, where the values of hidden variables $\{t^{(o)}\}$ are computed during the previous E-step. We can also impose a prior on each parameter. The prior probabilities of $\alpha_0^{(k)}$, $\alpha_1^{(k)}$, and $\beta^{(o)}$ are denoted as $p(\alpha_0^{(k)})$, $p(\alpha_1^{(k)})$, and $p(\beta^{(o)})$, respectively.

$$
\begin{aligned}
Q(\{\boldsymbol{\alpha}^{(k)}\},\{\beta^{(o)}\}) &= \mathbb{E}\big[\ln\big(p(L,\{t^{(o)}\}|\{\boldsymbol{\alpha}^{(k)}\},\{\beta^{(o)}\})p(\{\boldsymbol{\alpha}^{(k)}\},\{\beta^{(o)}\})\big)\big] \\
&= \mathbb{E}\big[\ln\prod_{(o)}\big(p(t^{(o)})\prod_{(k)\in K^{(o)}} p(l_{(o)}^{(k)}|t^{(o)},\alpha_{t^{(o)}}^{(k)},\beta^{(o)})\big)\big] \\
&\quad + \ln\prod_{(k)}\prod_{i=0}^{1} p(\alpha_i^{(k)}) + \ln\prod_{(o)} p(\beta^{(o)}) \\
&= \sum_{(o)}\mathbb{E}\big[\ln p(t^{(o)})\big] + \sum_{(o)}\sum_{(k)\in K^{(o)}}\mathbb{E}\big[\ln p(l_{(o)}^{(k)}|t^{(o)},\alpha_{t^{(o)}}^{(k)},\beta^{(o)})\big] \\
&\quad + \sum_{(k)}\sum_{i=0}^{1}\ln p(\alpha_i^{(k)}) + \sum_{(o)}\ln p(\beta^{(o)})
\end{aligned}
\tag{9}
$$

where the expectation is taken over $\{t^{(o)}\}$ estimated during the previous E-step. The values of $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\beta^{(o)}\}$ are obtained by maximizing the auxiliary function Q. This is not directly solvable, therefore we

apply a gradient ascent approach to find parameter values that locally maximize Q.

Let us define $p_i^{(o)} = p(t^{(o)} = i)$ estimated from the previous E-step, then

$$Q(\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\}) = \sum_{(o)} \sum_{i=0}^{1} p_i^{(o)} \ln p(t^{(o)} = i) + \sum_{(o)} \sum_{(k)\in K^{(o)}} \sum_{i=0}^{1} p_i^{(o)} \ln p(l_{(o)}^{(k)} | t^{(o)} = i, \alpha_i^{(k)}, \beta^{(o)}) + \sum_{(k)} \sum_{i=0}^{1} \ln p(\alpha_i^{(k)}) + \sum_{(o)} \ln p(\beta^{(o)})$$

Based on equation (1) and (2), we have:

$$p(l_{(o)}^{(k)} | t^{(o)} = 0, \alpha_0^{(k)}, \beta^{(o)}) = \sigma(\alpha_0^{(k)}\beta^{(o)})^{1-l_{(o)}^{(k)}} \big(1 - \sigma(\alpha_0^{(k)}\beta^{(o)})\big)^{l_{(o)}^{(k)}}$$

and

$$p(l_{(o)}^{(k)} | t^{(o)} = 1, \alpha_1^{(k)}, \beta^{(o)}) = \sigma(\alpha_1^{(k)}\beta^{(o)})^{l_{(o)}^{(k)}} \big(1 - \sigma(\alpha_1^{(k)}\beta^{(o)})\big)^{1-l_{(o)}^{(k)}}$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic function. Then,

$$\begin{aligned} Q(\{\boldsymbol{\alpha}^{(k)}\}, \{\beta^{(o)}\}) = & \sum_{(o)} \big(p_0^{(o)} \ln p(t^{(o)} = 0) + p_1^{(o)} \ln p(t^{(o)} = 1)\big) \\ & + \sum_{(o)} \sum_{(k)\in K^{(o)}} p_0^{(o)} \big((1 - l_{(o)}^{(k)}) \ln \sigma(\alpha_0^{(k)}\beta^{(o)}) + l_{(o)}^{(k)} \ln(1 - \sigma(\alpha_0^{(k)}\beta^{(o)}))\big) \\ & + \sum_{(o)} \sum_{(k)\in K^{(o)}} p_1^{(o)} \big(l_{(o)}^{(k)} \ln \sigma(\alpha_1^{(k)}\beta^{(o)}) + (1 - l_{(o)}^{(k)}) \ln(1 - \sigma(\alpha_1^{(k)}\beta^{(o)}))\big) \\ & + \sum_{(k)} \big(\ln p(\alpha_0^{(k)}) + \ln p(\alpha_1^{(k)})\big) + \sum_{(o)} \ln p(\beta^{(o)}) \end{aligned}$$

Using the fact that

$$\frac{d}{dx} \ln \sigma(x) = 1 - \sigma(x)$$

and

$$\frac{d}{dx} \ln(1 - \sigma(x)) = -\sigma(x)$$

we differentiate function Q with respect to $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\beta^{(o)}\}$:

$$\begin{aligned} \frac{\partial Q}{\partial \alpha_0^{(k)}} &= \sum_{(o)\in O^{(k)}} p_0^{(o)} \big((1 - l_{(o)}^{(k)})(1 - \sigma(\alpha_0^{(k)}\beta^{(o)}))\beta^{(o)} - l_{(o)}^{(k)} \sigma(\alpha_0^{(k)}\beta^{(o)})\beta^{(o)}\big) + \frac{d \ln p(\alpha_0^{(k)})}{d\alpha_0^{(k)}} \\ &= \sum_{(o)\in O^{(k)}} p_0^{(o)} \beta^{(o)} \big(1 - l_{(o)}^{(k)} - \sigma(\alpha_0^{(k)}\beta^{(o)})\big) + \frac{d \ln p(\alpha_0^{(k)})}{d\alpha_0^{(k)}} \end{aligned}$$

$$\frac{\partial Q}{\partial \alpha_1^{(k)}} = \sum_{(o)\in O^{(k)}} p_1^{(o)} \big(l_{(o)}^{(k)} (1-\sigma(\alpha_1^{(k)}\beta^{(o)}))\beta^{(o)} - (1-l_{(o)}^{(k)})\sigma(\alpha_1^{(k)}\beta^{(o)})\beta^{(o)} \big) + \frac{d\ln p(\alpha_1^{(k)})}{d\alpha_1^{(k)}}$$
$$= \sum_{(o)\in O^{(k)}} p_1^{(o)}\beta^{(o)} \big(l_{(o)}^{(k)} - \sigma(\alpha_1^{(k)}\beta^{(o)}) \big) + \frac{d\ln p(\alpha_1^{(k)})}{d\alpha_1^{(k)}}$$

$$\frac{\partial Q}{\partial \beta^{(o)}} = \sum_{(k)\in K^{(o)}} p_0^{(o)}\alpha_0^{(k)} \big(1 - l_{(o)}^{(k)} - \sigma(\alpha_0^{(k)}\beta^{(o)}) \big) + p_1^{(o)}\alpha_1^{(k)} \big(l_{(o)}^{(k)} - \sigma(\alpha_1^{(k)}\beta^{(o)}) \big) + \frac{d\ln p(\beta^{(o)})}{d\beta^{(o)}}$$

To find locally optimal values of $\{\boldsymbol{\alpha}^{(k)}\}$ and $\{\beta^{(o)}\}$, we set the gradient to zero. The resulting equations are non-linear and we use iterative methods to solve them. Using gradient ascent, we take steps proportional to the positive of the gradient and approach the local maximum of the function eventually.

Appendix C Proofs

C.1 Proof of Proposition 3

Proof: Proof. The estimated misclassification cost at step m is

$$EstCost^{(o)}|_m = EstCost(\mathbf{p}^{(o)}|_m) = \min_{j\in\{0,1\}} \sum_{i=0}^{1} p_i^{(o)}|_m c_{ij} = \min\{p_1^{(o)}|_m c_{10}, p_0^{(o)}|_m c_{01}\}$$

Worker (k) assigns to object (o) a label 0 with probability $p(l_{(o)}^{(k)} = 0) = p_0^{(o)}|_m \xi_{00}^{(k,o)} + p_1^{(o)}|_m \xi_{10}^{(k,o)}$, and a label 1 with probability $p(l_{(o)}^{(k)} = 1) = p_0^{(o)}|_m \xi_{01}^{(k,o)} + p_1^{(o)}|_m \xi_{11}^{(k,o)}$, where $\xi_{00}^{(k,o)} = \frac{1}{1+e^{-\alpha_0^{(k)}\beta^{(o)}}}$, $\xi_{10}^{(k,o)} = \frac{1}{1+e^{\alpha_1^{(k)}\beta^{(o)}}}$, $\xi_{01}^{(k,o)} = \frac{1}{1+e^{\alpha_0^{(k)}\beta^{(o)}}}$, and $\xi_{11}^{(k,o)} = \frac{1}{1+e^{-\alpha_1^{(k)}\beta^{(o)}}}$.

If $l_{(o)}^{(k)} = 0$, the new class probability estimate for object (o) is

$$\mathbf{p}^{(o)}|_{(m+1)}^{0} = \Big(\frac{p_0^{(o)}|_m \xi_{00}^{(k,o)}}{p_0^{(o)}|_m \xi_{00}^{(k,o)} + p_1^{(o)}|_m \xi_{10}^{(k,o)}}, \frac{p_1^{(o)}|_m \xi_{10}^{(k,o)}}{p_0^{(o)}|_m \xi_{00}^{(k,o)} + p_1^{(o)}|_m \xi_{10}^{(k,o)}}\Big)$$

and the associated estimated misclassification cost is

$$\begin{aligned} EstCost(\mathbf{p}^{(o)}|_{(m+1)}^{0}) &= \min\{p_1^{(o)}|_{(m+1)}^{0} c_{10}, p_0^{(o)}|_{(m+1)}^{0} c_{01}\} \\ &= \min\Big\{\frac{p_1^{(o)}|_m \xi_{10}^{(k,o)}}{p_0^{(o)}|_m \xi_{00}^{(k,o)} + p_1^{(o)}|_m \xi_{10}^{(k,o)}} c_{10}, \frac{p_0^{(o)}|_m \xi_{00}^{(k,o)}}{p_0^{(o)}|_m \xi_{00}^{(k,o)} + p_1^{(o)}|_m \xi_{10}^{(k,o)}} c_{01}\Big\} \end{aligned}$$

If $l_{(o)}^{(k)} = 1$, the new class probability estimate for object (o) is

$$\mathbf{p}^{(o)}|_{(m+1)}^{1} = \Big(\frac{p_0^{(o)}|_m \xi_{01}^{(k,o)}}{p_0^{(o)}|_m \xi_{01}^{(k,o)} + p_1^{(o)}|_m \xi_{11}^{(k,o)}}, \frac{p_1^{(o)}|_m \xi_{11}^{(k,o)}}{p_0^{(o)}|_m \xi_{01}^{(k,o)} + p_1^{(o)}|_m \xi_{11}^{(k,o)}}\Big)$$

and the associated estimated misclassification cost is

$$\begin{aligned} EstCost(\mathbf{p}^{(o)}|_{(m+1)}^{1}) &= \min\{p_1^{(o)}|_{(m+1)}^{1} c_{10}, p_0^{(o)}|_{(m+1)}^{1} c_{01}\} \\ &= \min\Big\{\frac{p_1^{(o)}|_m \xi_{11}^{(k,o)}}{p_0^{(o)}|_m \xi_{01}^{(k,o)} + p_1^{(o)}|_m \xi_{11}^{(k,o)}} c_{10}, \frac{p_0^{(o)}|_m \xi_{01}^{(k,o)}}{p_0^{(o)}|_m \xi_{01}^{(k,o)} + p_1^{(o)}|_m \xi_{11}^{(k,o)}} c_{01}\Big\} \end{aligned}$$

Therefore, the expected misclassification cost at step $(m+1)$ is

$$\begin{aligned}\mathbb{E}(EstCost^{(o)}|_{(m+1)}) &= p(l_{(o)}^{(k)}=0)EstCost(\mathbf{p}^{(o)}|_{(m+1)}^{0}) + p(l_{(o)}^{(k)}=1)EstCost(\mathbf{p}^{(o)}|_{(m+1)}^{1}) \\ &= (p_0^{(o)}|_m\xi_{00}^{(k,o)} + p_1^{(o)}|_m\xi_{10}^{(k,o)})\min\Big\{\frac{p_1^{(o)}|_m\xi_{10}^{(k,o)}}{p_0^{(o)}|_m\xi_{00}^{(k,o)} + p_1^{(o)}|_m\xi_{10}^{(k,o)}}c_{10}, \frac{p_0^{(o)}|_m\xi_{00}^{(k,o)}}{p_0^{(o)}|_m\xi_{00}^{(k,o)} + p_1^{(o)}|_m\xi_{10}^{(k,o)}}c_{01}\Big\} \\ &\quad + (p_0^{(o)}|_m\xi_{01}^{(k,o)} + p_1^{(o)}|_m\xi_{11}^{(k,o)})\min\Big\{\frac{p_1^{(o)}|_m\xi_{11}^{(k,o)}}{p_0^{(o)}|_m\xi_{01}^{(k,o)} + p_1^{(o)}|_m\xi_{11}^{(k,o)}}c_{10}, \frac{p_0^{(o)}|_m\xi_{01}^{(k,o)}}{p_0^{(o)}|_m\xi_{01}^{(k,o)} + p_1^{(o)}|_m\xi_{11}^{(k,o)}}c_{01}\Big\} \\ &= \min\{p_1^{(o)}|_m\xi_{10}^{(k,o)}c_{10}, p_0^{(o)}|_m\xi_{00}^{(k,o)}c_{01}\} + \min\{p_1^{(o)}|_m\xi_{11}^{(k,o)}c_{10}, p_0^{(o)}|_m\xi_{01}^{(k,o)}c_{01}\}\end{aligned}$$

If the predicted label at step m and step $(m+1)$ is 0, then:

$$\mathbb{E}(EstCost^{(o)}|_{(m+1)}) = p_1^{(o)}|_m\xi_{10}^{(k,o)}c_{10} + p_1^{(o)}|_m\xi_{11}^{(k,o)}c_{10} = p_1^{(o)}|_m c_{10}(\xi_{10}^{(k,o)} + \xi_{11}^{(k,o)}) = p_1^{(o)}|_m c_{10} = EstCost^{(o)}|_m$$

If the predicted label at step m and step $(m+1)$ is 1, then:

$$\mathbb{E}(EstCost^{(o)}|_{(m+1)}) = p_0^{(o)}|_m\xi_{00}^{(k,o)}c_{01} + p_0^{(o)}|_m\xi_{01}^{(k,o)}c_{01} = p_0^{(o)}|_m c_{01}(\xi_{00}^{(k,o)} + \xi_{01}^{(k,o)}) = p_0^{(o)}|_m c_{01} = EstCost^{(o)}|_m$$

Therefore, $EstCost^{(o)}|_m = \mathbb{E}(EstCost^{(o)}|_{(m+1)})$. □ □

C.2 Proof of Proposition 4

Proof: Proof. The estimated misclassification cost at step m is

$$EstCost^{(o)}|_m = EstCost(\mathbf{p}^{(o)}|_m) = \min_{j\in\{0,1\}}\sum_{i=0}^{1} p_i^{(o)}|_m c_{ij} = \min\{p_1^{(o)}|_m c_{10}, p_0^{(o)}|_m c_{01}\}$$

Worker (k) assigns to object (o) a label 0 with probability $p(l_{(o)}^{(k)}=0) = p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}$, and a label 1 with probability $p(l_{(o)}^{(k)}=1) = p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}$.

If $l_{(o)}^{(k)} = 0$, the new class probability estimate for object (o) is

$$\mathbf{p}^{(o)}|_{(m+1)}^{0} = \Big(\frac{p_0^{(o)}|_m e_{00}^{(k)}}{p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}}, \frac{p_1^{(o)}|_m e_{10}^{(k)}}{p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}}\Big)$$

and the associated estimated misclassification cost is

$$\begin{aligned}EstCost(\mathbf{p}^{(o)}|_{(m+1)}^{0}) &= \min\{p_1^{(o)}|_{(m+1)}^{0}c_{10}, p_0^{(o)}|_{(m+1)}^{0}c_{01}\} \\ &= \min\Big\{\frac{p_1^{(o)}|_m e_{10}^{(k)}}{p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}}c_{10}, \frac{p_0^{(o)}|_m e_{00}^{(k)}}{p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}}c_{01}\Big\}\end{aligned}$$

If $l_{(o)}^{(k)} = 1$, the new class probability estimate for object (o) is

$$\mathbf{p}^{(o)}|_{(m+1)}^{1} = \left(\frac{p_0^{(o)}|_m e_{01}^{(k)}}{p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}}, \frac{p_1^{(o)}|_m e_{11}^{(k)}}{p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}}\right)$$

and the associated estimated misclassification cost is

$$\begin{aligned} EstCost(\mathbf{p}^{(o)}|_{(m+1)}^{1}) &= \min\{p_1^{(o)}|_{(m+1)}^{1} c_{10}, p_0^{(o)}|_{(m+1)}^{1} c_{01}\} \\ &= \min\left\{\frac{p_1^{(o)}|_m e_{11}^{(k)}}{p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}} c_{10}, \frac{p_0^{(o)}|_m e_{01}^{(k)}}{p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}} c_{01}\right\} \end{aligned}$$

Therefore, the expected variation in misclassification cost is

$$\begin{aligned} &\mathbb{E}\left(\left|EstCost^{(o)}|_m - EstCost^{(o)}|_{(m+1)}\right|\right) \\ &= p(l_{(o)}^{(k)} = 0)\left|EstCost^{(o)}|_m - EstCost^{(o)}|_{(m+1)}^{0}\right| + p(l_{(o)}^{(k)} = 1)\left|EstCost^{(o)}|_m - EstCost^{(o)}|_{(m+1)}^{1}\right| \\ &= (p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)})\left|\min\{p_1^{(o)}|_m c_{10}, p_0^{(o)}|_m c_{01}\} - \min\{\frac{p_1^{(o)}|_m e_{10}^{(k)}}{p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}} c_{10}, \frac{p_0^{(o)}|_m e_{00}^{(k)}}{p_0^{(o)}|_m e_{00}^{(k)} + p_1^{(o)}|_m e_{10}^{(k)}} c_{01}\}\right| \\ &\quad + (p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)})\left|\min\{p_1^{(o)}|_m c_{10}, p_0^{(o)}|_m c_{01}\} - \min\{\frac{p_1^{(o)}|_m e_{11}^{(k)}}{p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}} c_{10}, \frac{p_0^{(o)}|_m e_{01}^{(k)}}{p_0^{(o)}|_m e_{01}^{(k)} + p_1^{(o)}|_m e_{11}^{(k)}} c_{01}\}\right| \end{aligned}$$

The value of the above function only depends on the class probability estimate $\mathbf{p}^{(o)}|_m$, the confusion matrix of the worker $\mathbf{e}^{(k)}$, and the cost matrix $\mathbf{c}$. □ □

Appendix D Simulation Results: Inference Algorithms in a Static System

D.1 Object Actual Misclassification Cost

Since true classes are known, we can calculate the actual misclassification cost of each object under different inference algorithms based on Proposition 2. We report the average actual misclassification cost of objects as a function of the average number of labels assigned per object. The results in Figure D1(a) are obtained under the symmetric cost matrix $\mathbf{c}^{(a)}$, and the results in Figure D1(b) are obtained under the asymmetric cost matrix $\mathbf{c}^{(b)}$.

We see that under both cost specifications, EM and GLAD outperform MV and MP consistently. The performance gap becomes more pronounced when the cost matrix is asymmetric, which is not surprising since MV and MP only focus on prediction error rate, while EM and GLAD take into account the costs associated with different types of classification errors when making predictions.

It is worth noting that GLAD achieves similar performance as EM when the cost matrix is symmetric, but possesses a clear advantage over EM when the cost matrix is asymmetric. What causes the differential performance between GLAD and EM when the cost matrix is asymmetric? We turn to the basic assumption underlying EM algorithm, that is, workers' error rates do not change when labeling objects of varying degrees of easiness. The consequence is that EM is likely to produce overconfident (or extreme) class probability estimates for difficult objects (See Appendix E for an explanation). The overconfident estimates may not change the label prediction in symmetric cost setting but have an impact on the label prediction in asymmetric cost setting. For instance, if the class probability estimates for an object are $(0.8, 0.2)$ using GLAD and $(0.9, 0.1)$ using EM, when the cost matrix is $\mathbf{c}^{(a)}$, both EM and GLAD report 0; however, when the cost matrix is $\mathbf{c}^{(b)}$, GLAD reports 1 but EM reports 0. By incorporating object easiness into inference, GLAD allows the employer to obtain more accurate class probability estimates for each object, yielding considerable improvements in cost reduction when facing asymmetric misclassification costs.

D.2 Worker Quality Estimation Accuracy

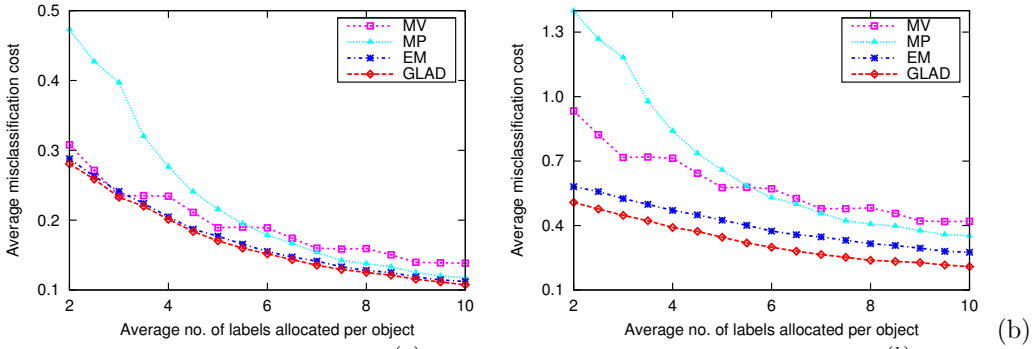
Following the notations introduced in Section 4, the quality measures for worker (k) using MV, MP, EM and GLAD are accuracy rate $q^{(k)}$, sum of worker messages $y_{(k)} = \sum_{(o)\in O^{(k)}} (2l_{(o)}^{(k)} - 1)x_{(o)\to(k)}$, confusion matrix $\mathbf{e}^{(k)}$, and quality vector $\hat{\boldsymbol{\alpha}}^{(k)}$, respectively. Since these measures are all at different scales and hard to compare directly, we resort to Spearman's rank correlation coefficient which provides a nonparametric estimate of the strength of association between two ranked variables. Table D1 shows how we calculate the Spearman correlation for each inference algorithm, where $\rho_{X,Y}$ denotes the Spearman's rho coefficient

between X and Y.

Algorithm	Quality Measure	Spearman Correlation
MV	Accuracy rate $q^{(k)}$	$0.5\rho_{\alpha_0^{(k)},q^{(k)}} + 0.5\rho_{\alpha_1^{(k)},q^{(k)}}$
MP	Sum of worker messages $y_{(k)}$	$0.5\rho_{\alpha_0^{(k)},y^{(k)}} + 0.5\rho_{\alpha_1^{(k)},y^{(k)}}$
EM	Confusion matrix $\mathbf{e}^{(k)}$	$0.5\rho_{\alpha_0^{(k)},e_{00}^{(k)}} + 0.5\rho_{\alpha_1^{(k)},e_{11}^{(k)}}$
GLAD	Quality vector $\hat{\boldsymbol{\alpha}}^{(k)}$	$0.5\rho_{\alpha_0^{(k)},\hat{\alpha}_0^{(k)}} + 0.5\rho_{\alpha_1^{(k)},\hat{\alpha}_1^{(k)}}$

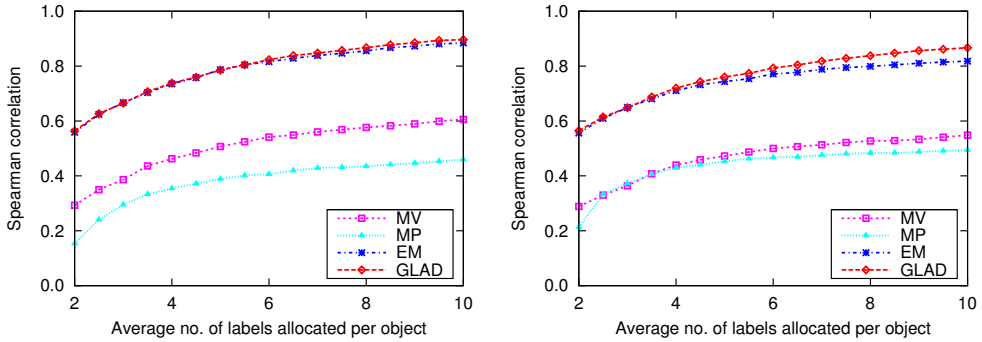
Table D1: Calculating the Spearman correlation for different inference algorithms

The correlation results obtained using different inference algorithms are presented in Figure D2(a). Contrary to our expectation, GLAD does not exhibit superior performance over EM in estimating worker quality. We attribute this to the uniform assignment of workers in the simulation. Under the uniform assignment, each worker is likely to be assigned with a similar mixture of easy ($\beta^{(o)} > 1$) and difficult ($\beta^{(o)} < 1$) objects. Since worker quality is computed by aggregating over all objects one has labeled, on average, EM won't over- or under-estimate the quality of workers. We then turn to a non-uniform assignment setting, in which some workers are disproportionately assigned with more easy (or difficult) objects. Specifically, we split the worker population into two halves: the first half is assigned with 75% easy objects and 25% difficult objects while the second half is assigned with 25% easy objects and 75% difficult objects. The correlation results obtained under this non-uniform assignment are reported in Figure D2(b), which demonstrates a slight advantage of GLAD over EM when the number of labels assigned to each object is relatively high. As more labels are collected, the easiness estimates of the objects using GLAD become more accurate, leading to a more fair evaluation of worker quality.



(a) symmetric cost matrix $\mathbf{c}^{(a)}$ asymmetric cost matrix $\mathbf{c}^{(b)}$ (b)

Figure D1: Average actual misclassification cost as a function of the average number of labels assigned per object for different inference algorithms in a static system



(a) uniform assignment (b) non-uniform assignment

Figure D2: Spearman correlation between worker quality estimates and true quality values as a function of the average number of labels assigned per object for different inference algorithms in a static system

Appendix E EM Produces Overconfident Estimates for Difficult Objects

For illustration purpose, we consider a very simple case where all the workers have homogenous labeling quality and the following relationship holds: $\alpha_0 = \alpha_1 > 0$. As the object receives more labels and worker quality estimates become more accurate, we will have $e_{00} \approx e_{11} > 0.5$ using EM and $\hat{\alpha}_0 \approx \hat{\alpha}_1 > 0$ using GLAD.

Under GLAD, let us denote $\hat{\xi}_{ii}^{(o)} = \frac{1}{1+e^{-\hat{\alpha}_i \hat{\beta}^{(o)}}}$. Since $\hat{\alpha}_i > 0$, $\hat{\xi}_{ii}^{(o)}$ decreases as $\hat{\beta}^{(o)}$ is getting smaller (i.e., the object (o) is more difficult). However, under EM, e_{ii} is the same across all the objects. When the object (o) is sufficiently difficult, the following relationship $\hat{\xi}_{ii}^{(o)} < e_{ii}$ holds.

Suppose that the object (o) has collected p positive labels and n negative labels.

For EM, we have

$$\begin{aligned}\mathbf{p}_{EM}^{(o)} &= \Big(\frac{(e_{00})^p(1-e_{00})^n}{(e_{00})^p(1-e_{00})^n + (1-e_{11})^p(e_{11})^n}, \frac{(1-e_{11})^p(e_{11})^n}{(e_{00})^p(1-e_{00})^n + (1-e_{11})^p(e_{11})^n}\Big) \\ &\approx \Big(\frac{1}{1+(\frac{1-e_{00}}{e_{00}})^{p-n}}, \frac{1}{1+(\frac{e_{00}}{1-e_{00}})^{p-n}}\Big)\end{aligned}$$

For GLAD, we have

$$\begin{aligned}\mathbf{p}_{GLAD}^{(o)} &= \Big(\frac{(\hat{\xi}_{00})^p(1-\hat{\xi}_{00})^n}{(\hat{\xi}_{00})^p(1-\hat{\xi}_{00})^n + (1-\hat{\xi}_{11})^p(\hat{\xi}_{11})^n}, \frac{(1-\hat{\xi}_{11})^p(\hat{\xi}_{11})^n}{(\hat{\xi}_{00})^p(1-\hat{\xi}_{00})^n + (1-\hat{\xi}_{11})^p(\hat{\xi}_{11})^n}\Big) \\ &\approx \Big(\frac{1}{1+(\frac{1-\hat{\xi}_{00}}{\hat{\xi}_{00}})^{p-n}}, \frac{1}{1+(\frac{\hat{\xi}_{00}}{1-\hat{\xi}_{00}})^{p-n}}\Big)\end{aligned}$$

Without loss of generality, we assume that $p > n$. Then the EM probability estimate of the most likely class 0 is $\frac{1}{1+(\frac{1-e_{00}}{e_{00}})^{p-n}}$, and the GLAD probability estimate of the most likely class 0 is $\frac{1}{1+(\frac{1-\hat{\xi}_{00}}{\hat{\xi}_{00}})^{p-n}}$. Since $\hat{\xi}_{00}^{(o)} < e_{00}$ holds when object (o) is sufficiently difficult, we have $\frac{1}{1+(\frac{1-e_{00}}{e_{00}})^{p-n}} > \frac{1}{1+(\frac{1-\hat{\xi}_{00}}{\hat{\xi}_{00}})^{p-n}}$. Therefore, the EM produces more confident class probability estimates for object (o).

To confirm this is indeed the case, we plot the probability estimates of most likely classes (i.e., $\max\{p_0^{(o)}, p_1^{(o)}\}$) by EM and GLAD for the top 10% most difficult objects in Figure E1,[24] which clearly shows that EM estimates are much more extreme (i.e., close to 1) than GLAD estimates.

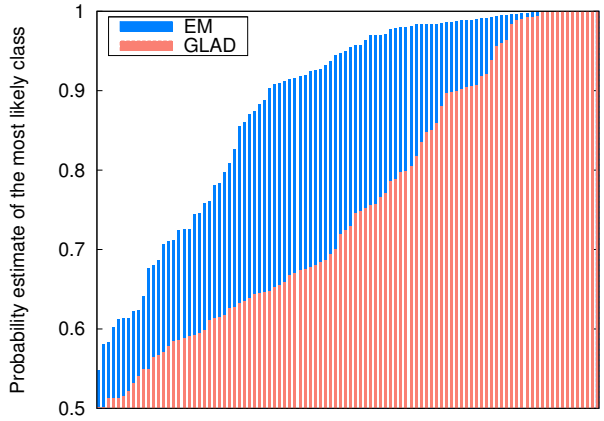
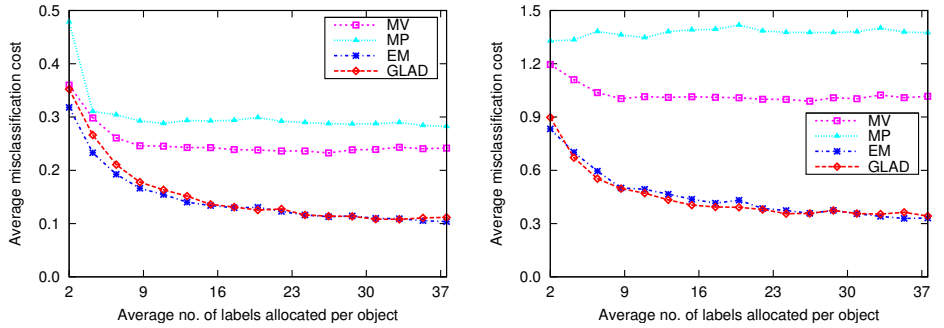


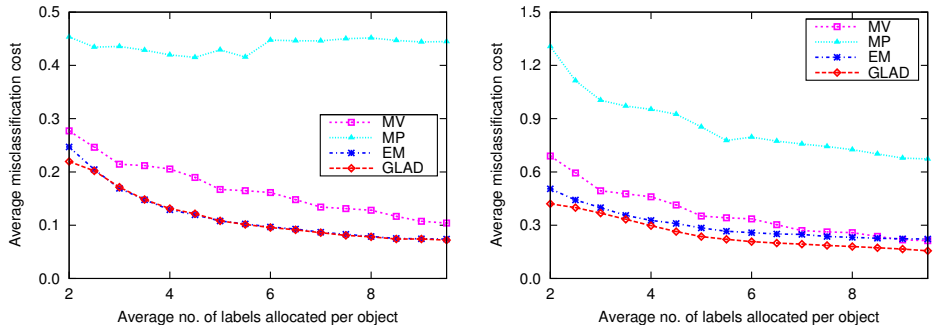
Figure E1: The probability estimates of most likely classes for the top 10% most difficult objects

[24]The results are obtained under the simulation setting in Section 6.

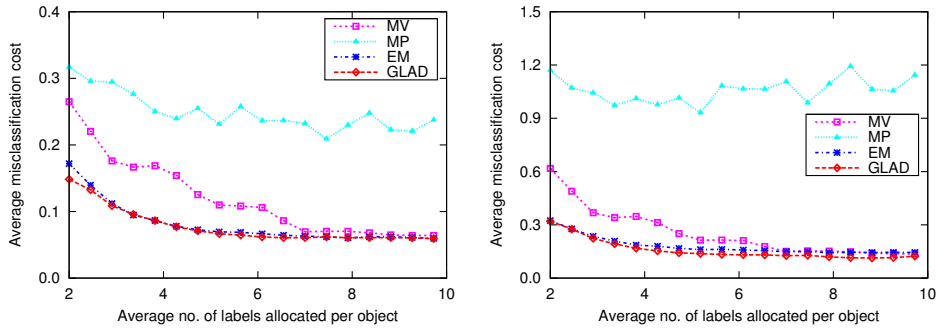
Appendix F Supplementary Figures



(a) *bluebird* dataset, symmetric cost $\mathbf{c}^{(a)}$ (b) *bluebird* dataset, asymmetric cost $\mathbf{c}^{(b)}$



(c) *rte* dataset, symmetric cost $\mathbf{c}^{(a)}$ (d) *rte* dataset, asymmetric cost $\mathbf{c}^{(b)}$



(e) *temp* dataset, symmetric cost $\mathbf{c}^{(a)}$ (f) *temp* dataset, asymmetric cost $\mathbf{c}^{(b)}$

Figure F1: Average actual misclassification cost for different inference algorithms on real-world datasets

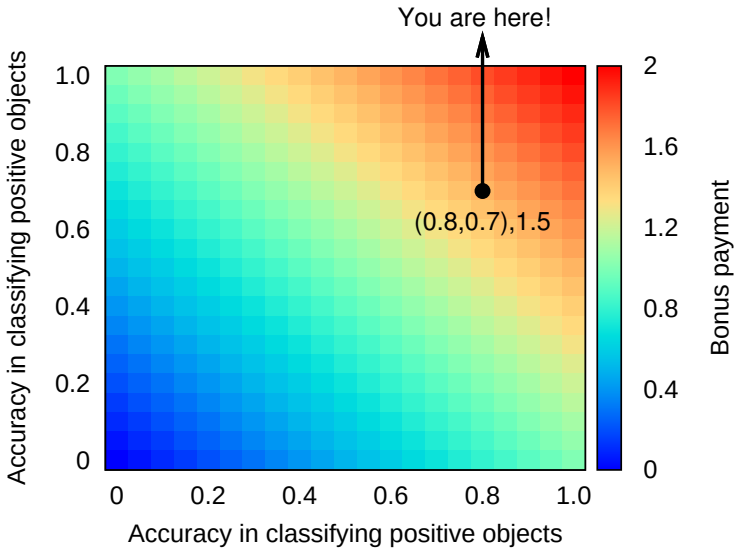


Figure F2: The interface of bonus payment to workers