Multicategory Crowdsourcing Accounting for Variable Task Difficulty, Worker Skill, and Worker Intention

Aditya Kurve, David J. Miller, *Senior Member, IEEE*, and George Kesidis

Abstract—Crowdsourcing allows instant recruitment of workers on the web to annotate image, webpage, or document databases. However, worker unreliability prevents taking a worker's responses at "face value". Thus, responses from multiple workers are typically aggregated to more reliably infer ground-truth answers. We study two approaches for crowd aggregation on multicategory answer spaces: stochastic modeling-based and deterministic objective function-based. Our stochastic model for answer generation plausibly captures the interplay between worker skills, intentions, and task difficulties and captures a broad range of worker types. Our deterministic objective-based approach aims to maximize the average aggregate confidence of weighted plurality crowd decision making. In both approaches, we explicitly model the *skill* and *intention* of individual workers, which is exploited for improved crowd aggregation. Our methods are applicable in both unsupervised and semi-supervised settings, and also when the batch of tasks is *heterogeneous*, i.e., from multiple domains, with task-dependent answer spaces. As observed experimentally, the proposed methods can defeat "tyranny of the masses", i.e., they are especially advantageous when there is an (a priori unknown) minority of skilled workers amongst a large crowd of unskilled (and malicious) workers.

Index Terms—Crowdsourcing, expectation-maximization, ensemble classification, inference, multicategory

1 Introduction

Crowdsourcing systems leverage diverse skill sets of a large number of Internet workers to solve problems and execute projects. In fact, the Linux project and Wikipedia can be considered products of crowdsourcing. These systems have recently gained much popularity with web services such as Amazon MTurk[1] and Crowd Flower,[2] which provide a convenient way for requestors to post problems to a large pool of online workers and get them solved quickly. The success of crowdsourcing has been demonstrated for annotating and labeling images and documents [1], writing and reviewing software code,[3] designing products [2], and also raising funds.[4] In this paper, we focus on tasks with categorical answer spaces.

Although the crowd expedites annotation, its anonymity allows noisy or even malicious labeling to occur. Even if malicious workers are not currently prevalent, identifying and countering them may become important in the future. Online reputation systems can help reduce the effect of noisy labels, but are susceptible to Sybil [3] or whitewashing [4] attacks. Moreover, the aggregate reputation score only reflects a worker's skill on *previous* tasks/domains. This may not be a good indication of his skill on new domains, for which he has not been evaluated. A second way to mitigate worker unreliability is to assign each task to multiple workers and aggregate their answers in some way to estimate the ground truth answer. The estimation may use simple voting or more sophisticated aggregation methods, e.g., [5], [6], [7]. The aggregation approaches we propose have several notable characteristics. First, we make a clear distinction between worker *skill* and *intention*, i.e., whether the worker is honest or malicious. This allows us to plausibly characterize the behavior of an adversarial worker in a multicategory setting. Our aggregation approaches explicitly identify such workers and exploit their behavior to, in fact, improve the crowd's accuracy (relative to the case of strictly non-malicious workers). Second, most approaches are only suitable for binary (two choice) tasks. By contrast, our approaches explicitly handle multicategory tasks. Beyond this, our approaches handle the case where the answer space is *task-dependent*, e.g., where some tasks in the batch under consideration may have four possible answers while others have seven. We believe few prior works have addressed this. Finally, some crowdsourcing methods [8], [7] exploit biases in the ground-truth answering mechanism or in the worker answering mechanism. Supposing the categorical answer space for all tasks is $\{a, b, c, d, e\}$, and that there is a *batch* of T tasks to be jointly solved by the crowd, several notable examples are:

1) A worker, when guessing, may, e.g., be more likely to choose the last answer (e), than the first (a).

1. www.mturk.com.
2. www.crowdflower.com.
3. www.topcoder.com.
4. www.crowdfunding.com.

- *A. Kurve and D.J. Miller are with the Department of Electrical Engineering, The Pennsylvania State University, PA 16802. E-mail: {ack205, djm25}@psu.edu.*
- *G. Kesidis is with the Department of Electrical Engineering and Computer Science and Engineering, The Pennsylvania State University, PA 16802. E-mail: gik2@psu.edu.*

Manuscript received 3 Sept. 2013; revised 1 Feb. 2014; accepted 11 May 2014. Date of publication 29 May 2014; date of current version 28 Jan. 2015. Recommended for acceptance by P. G. Ipeirotis. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2014.2327026

2) The ground-truth answer may be more likely to be, e.g., in position b than in position e.
3) All the tasks in the batch may come from the *same* classification domain, with answer a always corresponding to a particular class on the domain, b to a different class, and so on. If classes are not equally likely then this can be exploited.

Some methods exploit such biases. However, some of these biases could be easily removed, e.g., by randomly reordering the set of answers for every task in the batch. Moreover, considering Example 3, it is possible the batch consists of a mix of multiple domains. For example, workers with good visual acuity who are also world travelers may be given images and asked to identify country of origin for some, crop type for others, or whether there are vehicles present in a third subset. Each task in the batch may *not* be annotated by the underlying "domain". In this *heterogeneous* setting, answer a does *not* correspond to a particular category on a single domain. Thus, here, it is not possible for crowd aggregation to exploit single-domain, non-uniform class priors. When multiple domains are present, it is also likely the answer space cardinality is not fixed across all tasks.

The methods we develop do not attempt to exploit biases in the answer space (though our methods are easily extended to do so, when such biases are likely to be present). Instead, we solely exploit diversity in worker skill, task difficulty, and worker intention and our methods' capabilities to accurately infer these parameters, given a batch of tasks. Finally, many crowdsourcing approaches require proper setting of hyper-parameters, which is very difficult in an unsupervised setting (where there are no gold ("probe") tasks in the batch, for which the ground-truth answers are known). By contrast, our approaches do not require any hyper-parameters. We propose two distinct frameworks.

Our first approach assumes a stochastic model for answer generation that plausibly captures interplay between worker skills, intentions, and task difficulties. To the best of our knowledge, this is the first model that uses a common metric space (on the real line) to measure task difficulties and worker reliabilities, with the probability of answering correctly parameterized by their difference. In this model, we formalize the notion of an adversarial worker and model different types of adversaries. The simplest adversary is a random "spammer" who is equally likely to choose any of the possible answers. A somewhat "craftier" variant is a worker who gives incorrect answers "to the best of his skill level", i.e., who randomly chooses an incorrect answer when he knows the true answer, and reverts to pure random spamming when he does not know the true answer. It is interesting to investigate the effect of such workers on crowd accuracy, and to devise methods resilient to such workers, as malicious workers may disrupt crowdsourcing as well as opinion aggregation systems in general.

Detection of adversaries and estimation of worker skills and task difficulties can be assisted by the knowledge of ground-truth answers for some ("gold" or "probe"[5]) tasks. Accordingly, we formulate a semi-supervised approach, invoking a generalized Expectation-Maximization (GEM) algorithm [9] to maximize the joint log likelihood over the (known) true labels for the "probe" tasks and the answers of the crowd for all tasks. This specializes to an *unsupervised* method when no (labeled) probe tasks are available. Both cases are investigated experimentally in the sequel. Interestingly, our method's maximum a posteriori (MAP) crowd aggregation rule comes precisely from the E-step of our GEM algorithm, since the ground-truth answers are treated as the hidden data [10] in our GEM approach, with their expected values (the posterior probabilities) estimated in the E-step.

On some domains, our stochastic model may not well-characterize crowd behavior. Moreover, EM-based algorithms can be computationally burdensome. Accordingly, we also investigate approaches that do not assume underlying stochastic answer generation mechanisms and which are computationally lighter than our GEM approach. We propose two deterministic objective-based methods that jointly estimate worker intention, skill, and the ground truth answers by maximizing a measure of aggregate confidence on the estimated ground truth answers evaluated over the batch of tasks. Here, crowd aggregation is achieved by weighted plurality voting, with higher weights given to the workers estimated to be the most highly skilled and honest. These workers are identified essentially by their tendency to agree with each other on most of the tasks, unlike the low-skilled workers, who tend to answer arbitrarily.

Probe tasks are overhead, which limits the number of true tasks in a batch (of fixed size) that the crowd is solving. It may also be expensive, time-consuming and/or impractical to *devise* meaningful probe tasks. Accordingly, we consider our methods in both semi-supervised and *unsupervised* settings to evaluate the gains obtained by using probes. Our experimental evaluation of the proposed schemes consisted of three levels. First, we use simulated data generated in a way consistent with our proposed stochastic generative model. This allowed us to study robustness of our GEM algorithm by comparing estimated and actual model parameters. Second, we evaluated performance of the methods using a crowd of "simulated" workers that do not obviously generate answers in a fashion closely matched to our model. Specifically, each worker was a strong learner, formed as an ensemble of weak learners. Each weak learner was a decision tree, with the ensemble (and thus, a strong learner) obtained by multiclass boosting. A strong worker's skill was controlled by varying the number of boosting stages used. We performed experiments on UC Irvine data sets [11] and studied the comparative gains of our methods over benchmark methods on a variety of classification domains. We also performed experiments where the answer space (and its size) was not fixed across the tasks in the data batch. Our final experiment involved a crowdsourcing task we posted using Amazon Mturk. Overall, we observed our methods are especially advantageous compared to alternatives when there is an (a priori unknown) minority of skilled workers amongst a large crowd of unskilled (as well as malicious) workers, i.e., our methods overcome "tyranny of the masses".

5. The terminology "probe" comes from communications, where known symbols are transmitted through a channel, in order to learn the channel's characteristics.

2 Generative Semi-Supervised Modeling Approach

Here, we separately model worker intention and skill. We represent a worker's intention using a binary parameter indicating if he is adversarial or not. An honest worker provides accurate answers "to the best of his skill level" whereas an adversarial worker may provide incorrect answers "to the best of his skill level". In the case of binary crowdsourcing tasks, adversarial workers can be identified by a negative weight [6] given to their answers. Here we extend malicious/adversarial worker models to multicategory tasks and hypothesize both unsophisticated and somewhat "crafty" adversaries.

Our approach incorporates task difficulty and worker skill explicitly and, unlike previous approaches [1], [6], [12], characterizes their interplay. Task difficulty and worker skill are both represented on the real line, with our generative model for a worker's answer based on their difference. If the task difficulty exceeds a worker's skill level, the worker answers randomly (whether honest or adversarial). For an adversary, if the task difficulty is less than his skill level, he chooses randomly only from the set of incorrect answers. Another worker type is "spammers". These are lazy workers who simply answer randomly for all tasks. Our model identifies such workers by assigning them large negative skill values.

2.1 Notation

Suppose a crowd of N workers is presented with a set of T_u unlabeled tasks, for which the ground truth answers are unknown. There are also T_l probe tasks, with known ground truth answers.[6] We assume the crowd is unaware which tasks are probes. Accordingly, a malicious worker cannot alter his answering strategy in a customized way for the probe tasks to "fool" the system. Let $\{1, 2, \ldots, T_l\}$ be the index set of the probe tasks and $\{T_l + 1, T_l + 2, \ldots, T_l + T_u\}$ for the non-probe tasks. We assume without loss of generality that each worker is asked to solve all the tasks.[7] The answers for task i are chosen from an (in general) task-specific answer set $\mathcal{C}_i \equiv \{1, 2, \ldots, K_i\}$. Let $z_i \in \mathcal{C}_i$ be the ground truth answer. Let the response provided to the ith task by the jth worker be denoted $r_{ij} \in \mathcal{C}_i$. Our model's parameters are as follows: $\tilde{d}_i \in (-\infty, \infty)$ represents the difficulty level of task i; the intention of worker j is indicated by $v_j \in \{0, 1\}$, where $v_j = 1$ denotes an honest worker and $v_j = 0$ an adversary; $d_j \in (-\infty, \infty)$ represents the jth worker's (ground truth) skill level; and a_j is an additional parameter that controls an individual worker's propensity to answer correctly, for a given difference $d_j - \tilde{d}_i$.

2.2 Stochastic Generation Model

We define our model's *parameter set* as $\Lambda = \{\{(v_j, d_j, a_j) \forall j\}, \{\tilde{d}_i \forall i\}\}$. We hypothesize the generation of the answers for non-probe tasks in two steps. Independently, for each non-probe task $i \in \{T_l + 1, \ldots, T_l + T_u\}$:

(1) Randomly choose the ground truth answer (z_i) from $\mathcal{C}_i$ according to a uniform pmf.[8]
(2) For each worker $j \in \{1, \ldots, N\}$, generate $r_{ij} \in \mathcal{C}_i$ for task i based on the *parameterized* probability mass function (pmf) $\beta(r_{ij} \mid \Lambda_{ij}, z_i)$, where $\Lambda_{ij} := \{v_j, d_j, a_j, \tilde{d}_i\}$.[9]

Also, independently for each probe task $i \in \{1, \ldots, T_l\}$ and for each worker j, generate the answer $r_{ij} \in \mathcal{C}_i$ based on the parameterized pmf $\beta(r_{ij} \mid \Lambda_{ij}, z_i)$.

2.3 Worker Types

We model the ability of a worker to solve the task correctly using a sigmoid function based on the difference between the task difficulty and the worker's skill,[10] i.e., the probability that worker j can solve task i is: $\frac{1}{1+e^{-a_j(d_j - \tilde{d}_i)}}$. Note that a_j captures worker individuality. It is also possible to tie this parameter, i.e., set $a_j = a, \forall j$.

2.3.1 Honest Workers

For an honest worker ($v_j = 1$), the pmf β is defined as:

$$\beta(r_{ij} = l | \Lambda_{ij}, v_j = 1, z_i) = \begin{cases} \frac{1}{1+e^{-a_j(d_j - \tilde{d}_i)}} + \left(\frac{1}{K_i}\right)\left(\frac{e^{-a_j(d_j - \tilde{d}_i)}}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \text{for } l = z_i \\ \left(\frac{1}{K_i}\right)\left(\frac{e^{-a_j(d_j - \tilde{d}_i)}}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \text{otherwise.} \end{cases} \quad (1)$$

Here, the worker answers correctly with high probability if $d_j > \tilde{d}_i$, and with probability $\frac{1}{K_i}$ otherwise. Note that "spammer" workers can be represented by choosing $d_j \ll \min_i \tilde{d}_i$. With this choice, the worker tends to answer randomly for all tasks. Next, we discuss a model for unsophisticated adversaries that is motivated by and represents perhaps the simplest departure from the random "spammer".

2.3.2 Simple Adversarial Workers

We model an unsophisticated adversary as follows:

$$\beta(r_{ij} = l | \Lambda_{ij}, v_j = 0, z_i) = \begin{cases} \left(\frac{1}{K_i}\right)\left(\frac{e^{-a_j(d_j - \tilde{d}_i)}}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \text{for } l = z_i \\ \left(\frac{1}{K_i}\right)\left(\frac{e^{-a_j(d_j - \tilde{d}_i)}}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \\ + \left(\frac{1}{K_i - 1}\right)\left(\frac{1}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & otherwise. \end{cases} \quad (2)$$

Here, if the worker knows the correct answer, he excludes it and randomly chooses from ("spams") amongst the

6. The unsupervised setting is a special case where $T_l = 0$.

7. We only make this assumption for notational simplicity. Our methodology applies generally to the setting where each worker solves only a subset of the tasks.

8. This assumes no answer bias. If such bias may exist, we can learn additional model parameters that capture a non-uniform prior over the possible answers.

9. The specific parametric dependence of β on Λ_{ij} will be introduced shortly.

10. Alternative (soft) generalized step functions could also in principle be used here.

remaining answers. Alternatively, if he does not know the correct answer, he reverts to pure random spamming over the full set of answers (including the correct one). A model for a "craftier" adversary, one who tries to evade detection as a malicious worker, will be given in Section 6.

2.4 Incomplete, Complete and Expected Complete Data Log Likelihood

The observed data $\mathcal{X} = \mathcal{R} \cup \mathcal{Z}_{\mathcal{L}}$ consists of the set $\mathcal{R}$ of answers given by the workers to all the tasks, i.e., $r_{ij}\ \forall\ i, j$ and the set $\mathcal{Z}_{\mathcal{L}} = \{z_i | i \in \{1, 2, \ldots, T_l\}\}$ of ground truth answers to the probe tasks. We express $\mathcal{R} = \mathcal{R}_L \cup \mathcal{R}_U$, i.e., the union of answers to probe tasks and non-probe tasks. We choose the hidden data [10] to be the ground truth answers to the non-probe tasks, i.e., $Z_i,\ \ i \in \{T_l + 1, \ldots, T_l + T_u\}$. Based on the stochastic model in Section 2.2, the incomplete data log-likelihood, which we seek to maximize in estimating Λ, is given by

$$
\begin{aligned}
log\ \mathcal{L}_{inc} &= log\ \mathrm{P}(\mathcal{R}, \mathcal{Z}_L \,|\, \Lambda) = log\ \mathrm{P}(\mathcal{R}_L, \mathcal{R}_U, \mathcal{Z}_L \,|\, \Lambda) \\
&= log\ \mathrm{P}(\mathcal{R}_L, \mathcal{Z}_L \,|\, \Lambda) + log\ \mathrm{P}(\mathcal{R}_U \,|\, \Lambda) \\
&= \sum_{i=1}^{T_l} \sum_{j=1}^{N} log \frac{1}{K_i} \beta(r_{ij} \,|\, \Lambda_{ij}, z_i) \\
&\quad + \sum_{i=T_l+1}^{T_l+T_u} \sum_{j=1}^{N} log \frac{1}{K_i} \sum_{k=1}^{K_i} \beta(r_{ij} \,|\, \Lambda_{ij}, Z_i = k) \qquad (3) \\
&\propto \sum_{i=1}^{T_l} \sum_{j=1}^{N} log\ \beta(r_{ij} \,|\, \Lambda_{ij}, z_i) \\
&\quad + \sum_{i=T_l+1}^{T_l+T_u} \sum_{j=1}^{N} log \sum_{k=1}^{K_i} \beta(r_{ij} \,|\, \Lambda_{ij}, Z_i = k).
\end{aligned}
$$

Treating Z_i, $i = 1, \ldots, T_l$ as the hidden data within the EM framework [10], the expected complete data log-likelihood, where the expectation is with respect to the pmf $\mathrm{P}(Z_i = k | \mathcal{X}, \Lambda)$, can be written as:

$$
\begin{aligned}
\mathbb{E}[log\ \mathcal{L}_c | \mathcal{X}, \Lambda] &\propto \sum_{i=1}^{T_l} \sum_{j=1}^{N} log\ \beta(r_{ij} \,|\, \Lambda_{ij}, z_i) \\
&\quad + \sum_{i=T_l+1}^{T_l+T_u} \sum_{j=1}^{N} \sum_{k=1}^{K_i} \\
&\qquad [\mathrm{P}(Z_i = k \,|\, \mathcal{X}, \Lambda) log\ \beta(r_{ij} \,|\, \Lambda_{ij}, Z_i = k)] \\
&= \sum_{i=1}^{T_l} \sum_{j:r_{ij}=z_i} [v_j\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 1, z_i = r_{ij})) \\
&\quad + (1 - v_j)\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 0, z_i = r_{ij}))] \\
&\quad + \sum_{i=1}^{T_l} \sum_{j:r_{ij}\neq z_i} [v_j\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 1, z_i \neq r_{ij})) \\
&\quad + (1 - v_j)\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 0, z_i \neq r_{ij}))]
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \sum_{i=T_l+1}^{T_l+T_u} \sum_{k=1}^{K_i} \sum_{j:r_{ij}=k} \mathrm{P}(Z_i = k) \cdot \\
&\qquad [v_j\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 1, Z_i = k)) \\
&\quad + (1 - v_j)\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 0, Z_i = k))] \\
&\quad + \sum_{i=T_l+1}^{T_l+T_u} \sum_{k=1}^{K_i} \sum_{j:r_{ij}\neq k} \mathrm{P}(Z_i = k) \cdot \\
&\qquad [v_j\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 1, Z_i \neq k)) \\
&\quad + (1 - v_j)\ log(\beta(r_{ij} \,|\, \Lambda_{ij}, v_j = 0, Z_i \neq k))].
\end{aligned}
$$

2.5 The Generalized EM (GEM) Algorithm

We formulate our algorithm using the above defined expected complete data log-likelihood. The EM algorithm ascends monotonically in $log\ \mathcal{L}_{inc}$ with each iteration of the E and M steps [10]. In the *expectation* step, one calculates the pmf $\mathrm{P}(Z_i = k | \mathcal{X}, \Lambda^t)$ using the current parameter values Λ^t, and in the *maximization* step, one computes $\Lambda^{t+1} = \mathrm{argmax}_{\Lambda}\ \mathbb{E}[log\ \mathcal{L}_c | \mathcal{X}, \Lambda^t]$.

E step. In the E-step we compute the expected value of $\mathcal{Z}_u = \{Z_i, i = T_l + 1, \ldots, T_l + T_u\}$ given the observed data $\mathcal{X}$ and the current parameter estimates Λ^t. Based on our assumed stochastic model (Section 3.2), with data for each task generated i.i.d, we have that $\mathrm{P}(\mathcal{Z}_u | \mathcal{X}, \Lambda^t) = \prod_{i=T_l+1}^{T_l+T_u} \mathrm{P}(Z_i = z_i | \mathcal{X}, \Lambda^t)$. Moreover, again based on the assumed stochastic model and applying Bayes' rule, we can derive the closed form expression for the posterior pmf in the E-step as:

$$
\mathrm{P}_i(Z_i = k | \mathcal{X}, \Lambda^t) = \frac{\prod_{j=1}^{N} \beta\big(r_{ij} \,|\, \Lambda_{ij}^t, Z_i = k\big)}{\sum_{l=1}^{K} \prod_{j=1}^{N} \beta\big(r_{ij} \,|\, \Lambda_{ij}^t, Z_i = l\big)}, \qquad (4)
$$
$$
\forall k, \forall i \in \{T_l + 1, \ldots, T_u + T_l\}.
$$

Generalized M step: In the M-step of EM, one maximizes the expected complete data log-likelihood with respect to the model parameters:

$$
\Lambda^{t+1} = arg\ \max_{\Lambda}\ \mathbb{E}[log\ \mathcal{L}_c(\Lambda) \,|\, \mathcal{X}, \Lambda]. \qquad (5)
$$

Since Λ consists of mixed (both continuous and discrete) parameters, with a particular parametric dependence, and with 2^N (honest, adversarial) crowd configurations, it is not practically feasible to find a closed form solution to (5) for our model. Instead, we use a *generalized* M-step approach [9], [13] to iteratively maximize over the two parameter subsets $\{v_j\ \forall\ j\}$, and $\{\{(d_j, a_j) \forall\ j\}, \{\tilde{d}_i\ \forall\ i\}\}$.

M1 Substep. Since (4) is an additive function of terms that each depend on a single variable v_j, we can find a closed form solution for $v_j\ \forall\ j$ given all other parameters fixed:

$$
\tilde{v}_j = \underset{v_j \in \{0,1\}}{\arg\max}\, \mathbb{E}(log \mathcal{L}_c(\{v_j\}) \,|\, \mathcal{X}_j, \tilde{\Lambda} \backslash \{v_j\}). \qquad (6)
$$

Here $\mathcal{X}_j$ is the set of answers provided by the jth worker and the ground truth answers for the probe tasks that he answered and $\tilde{\Lambda}$ is the result of the previous M2 substep.

M2 Substep: We maximize $\mathbb{E}[log\ \mathcal{L}_c(\Lambda\backslash\{v_j\})|\mathcal{X}, \{\tilde{v}_j\}]$ with respect to $\Lambda\backslash\{v_j\}$ given $\{\tilde{v}_j\}$ fixed from the previous M1 substep. For this, we use a gradient ascent algorithm which ensures monotonic increase in $log\ \mathcal{L}_{inc}$, but which may only find a local maximum, rather than a global maximum of $\mathbb{E}[log\ \mathcal{L}_c(\Lambda\backslash\{v_j\})|\mathcal{X}, \{\tilde{v}_j\}]$. Gradient ascent is performed until the relative change in the log-likelihood between gradient steps falls below a specified threshold. At termination, the result is stored in $\tilde{\Lambda}\backslash\{v_j\}$. The M1 and M2 substeps are applied alternately, iteratively, until the M2 stopping condition is met after the first gradient step is taken. Λ^{t+1} stores the result of the generalized M-step at termination.

GEM iterations are performed until the relative change in log-likelihood between two successive GEM iterations falls below a specified threshold. The convergence of the GEM scheme is guaranteed to a local extremum of the incomplete data log-likelihood function [10], [14].

2.6 Inference

Note that the E-step (4) computes the a posteriori probability that each of the answers is the correct one. Thus, after our GEM learning has converged, a maximum a posteriori decision rule applied to (4) gives our crowd-aggregated estimates of the true answers for all non-probe tasks. That is, at GEM convergence, the E-step directly gives our method's inference/decisionmaking.

2.7 Unsupervised GEM

Note that when probe tasks are not included in the batch, an unsupervised specialization of the above GEM algorithm is obtained. In particular, we have $T_l = 0$, with the first term in (3) and the first two terms in (4) not present. Our above GEM algorithm accordingly specializes. In Section 4, we will experimentally evaluate this unsupervised GEM scheme along with all other methods.

3 Energy-Constrained Weighted Plurality Aggregation

Performance of our GEM approach will in general depend on how well the true answer generation mechanism resembles the stochastic model assumed in Section 2.2. EM techniques are also well-known to be relatively computationally heavy. Thus, we also explore an alternative "principle" on which to base crowd aggregation, without an explicit stochastic model. The methods we propose in this section use weighted plurality voting, where the weights assigned to workers essentially reflect their individual skill level. A key idea here is to make the weight vector "energy-constrained", so as to obtain a bounded solution to the resulting optimization problem. The methods we will propose are applicable to both unsupervised and semi-supervised settings, but for clarity of presentation, we will focus on the unsupervised setting and then delineate an extension to exploit probe tasks, if available.

3.1 From Simple Plurality to Weighted Plurality Voting

Let $T = T_l + T_u$ and $\hat{\boldsymbol{z}}_i$ be the $K_i \times 1$ vector representing the inferred ground truth answers with $\hat{z}_{im} \in \{0, 1\}$ and $\sum_{m=1}^{K_i} \hat{z}_{im} = 1$, i.e., $\hat{z}_{im}$ is 1 when the inferred answer to the ith task is m. Also, $\hat{\boldsymbol{Z}} = (\hat{\boldsymbol{z}}_i, i = 1, \ldots, T)$. All other definitions from Section 2.1 will be used. A natural multicategory extension of majority voting is plurality voting, where the answer that gets the maximum number of votes is the inferred one. To help motivate what follows, we note that plurality voting is the solution of a maximization problem defined over a given batch of tasks. In particular it solves[11]:

$$\max_{\hat{\boldsymbol{Z}}} \sum_{i=1}^{T} \sum_{m=1}^{K_i} \hat{z}_{im} \sum_{j=1}^{N} \delta(r_{ij} - m) \quad subject\ to\ \hat{z}_{im} \in \{0, 1\},\ \sum_{m}^{K_i} \hat{z}_{im} = 1, i = 1, \ldots, T, \tag{7}$$

where $\sum_{j=1}^{N} \delta(r_{ij} - m)$ is the vote count for answer i.

Plurality-based voting is expected to perform well when all workers are honest and "equally reliable", with worker accuracy greater than that of a spammer.[12] However, the crowd may have workers with varying skill levels (and intentions), as considered before in Section 2. For the most challenging "tyranny of the masses" case where a small proportion of highly skilled workers exist among a mass of unskilled workers or spammers, standard plurality-based voting will be highly sub-optimal. Even supposing the highly skilled workers are *always* correct, "one worker, one vote" means that, if the skilled worker subset is a small minority, it will not in general be able to "tip the balance" of the plurality towards the correct answer. Alternatively, here we consider *weighted* voting schemes, where different weights are assigned to the workers based on their "accuracy level", accounting for both intention and skill. Allocation of higher weights to the most skilled workers may allow defeating "tyranny of the masses". Moreover, for weighted plurality voting, ties will almost never occur.

Thus, we propose maximizing a *weighted* plurality extension of (7): $\sum_{j=1}^{N} w_j \sum_{i=1}^{T} \sum_{m=1}^{K_i} \hat{z}_{im} \delta(r_{ij} - m)$. In order to avoid *unbounded* weight vector solutions, it is necessary to impose constraints on the weights of the workers. One possibility might seem to be to impose a constraint on the sum of the weights (e.g., that this sums to 1). However, maximizing the weighted plurality objective subject to this constraint yields a peculiar solution, with *all* the weight assigned to worker $j^* = \arg\ max_j \sum_{i=1}^{T} \sum_{m=1}^{K_i} \hat{z}_{im} \delta(r_{ij} - m)$. Clearly this solution does not exploit the full crowd and is undesirable. Alternatively, imposing a constraint on the sum of the *squared* weights ensures bounded solutions with *all* workers (except pure spammers, whose weights will be zero) participating in the decisionmaking. Also, for any given value of the energy constraint, all the weights in the solution (below) get scaled by a common factor. Thus, the weighted plurality answer remains the same, irrespective of the particular value chosen for the constraint. That is, the crowd-aggregated decisions are not affected by this choice—the *only* consequence of the constraint is ensuring

11. Ties could be broken by randomly selecting from among the set of plurality answers.

12. The expected accuracy of the spammer is $\frac{1}{K_i}$, where K_i is the number of possible answers for task i.

a bounded weight vector. Accordingly, we pose the following problem:

$$\max_{\hat{\boldsymbol{Z}},\boldsymbol{w}} \psi_{\text{wp}}(\boldsymbol{w},\hat{\boldsymbol{Z}}) = \sum_{j=1}^{N} w_j \sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}\delta(r_{ij}-m),$$
$$subject\ to \sum_{j=1}^{N} w_j^2 = 1, \hat{z}_{im} \in \{0,1\}, \sum_{m=1}^{K_i} \hat{z}_{im} = 1. \tag{8}$$

Since (8) is a nonconvex problem, we will solve it in a locally optimal fashion via an iterative algorithm, alternating between the updates of the weights and the inferred answers. We iterate over the following two (local maximization) steps until there are no further increases in the objective function.

Algorithm 1:

Step 1. For fixed $\boldsymbol{w}$ and for each task i, the choice for $\hat{\boldsymbol{z}}_i$ which maximizes (8) is:

$$\hat{z}_{im} = \begin{cases} 1 & \text{if } \sum_{j:r_{ij}=m} w_j > \sum_{j:r_{ij}=k} w_j \ \forall\, k \neq m \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Step 2. Given fixed $\hat{\boldsymbol{Z}}$ we compute the optimum $\boldsymbol{w}$, maximizing (8).The Lagrangian for this optimization problem is

$$\mathcal{L} = \sum_{j=1}^{N} w_j \sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}\delta(r_{ij}-m) + \lambda\left(\sum_{j=1}^{N} w_j^2 - 1\right). \tag{10}$$

Differentiating with respect to w_k, we get

$$\frac{\partial \mathcal{L}}{\partial w_k} = \sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}\delta(r_{ik}-m) + 2\lambda w_k = 0$$
$$\Rightarrow w_k = \frac{-1}{2\lambda}\sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}\delta(r_{ik}-m) \quad \forall k. \tag{11}$$

We can compute λ and find the optimal value of w_k as

$$w_k^* = \frac{\sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}(\delta(r_{ik}-m)}{\sqrt{\sum_{j=1}^{N}\left(\sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}\delta(r_{ij}-m)\right)^2}} \quad \forall k. \tag{12}$$

Each of the above two steps ascends in the objective (reward) function ψ_{wp}. We iterate between step 1 and step 2 until convergence (proved in the following).

Theorem 3.1. *Algorithm* 1 *monotonically increases in the objective (8), terminates in a finite number of iterations, and at a fixed point if there are no voting ties. Moreover, this fixed point is a local maximum of (8).*

The proof is given in Appendix B.

3.2 Accounting for Adversaries

Note that we have not accounted for adversarial workers in (8). To do so, suppose that an adversarial worker will choose the incorrect answer randomly (uniformly over all incorrect answers). In the following we will develop two extensions of the weight-constrained problem to accommodate the worker's intention. Our first approach uses binary parameters to represent worker intentions, whereas our second method exploits the extra degree of freedom provided by the *sign* of the estimated weight w_j to represent the worker's intention.

3.2.1. Introducing Binary Parameters to Represent Intention

Suppose we introduce an additional set of variables given by the $N \times 1$ vector $\boldsymbol{v}$, where $v_j \in \{0,1\}$ and where $v_j = 1$ and $v_j = 0$ characterize worker j as honest or adversarial, respectively. Accordingly, we rewrite the optimization problem as:

$$\max_{\hat{\boldsymbol{Z}},\boldsymbol{w},\boldsymbol{v}} \ \psi_{\text{bp}}(\boldsymbol{w},\hat{\boldsymbol{Z}},\boldsymbol{v}) = \sum_{j=1}^{N}\sum_{i=1}^{T}\sum_{m=1}^{K_i}\Big(v_j\hat{z}_{im}w_j\delta(r_{ij}-m) + \frac{1}{K_i-1}(1-v_j)\hat{z}_{im}w_j(1-\delta(r_{ij}-m))\Big),$$
$$subject\ to \sum_{j=1}^{N} w_j^2 = 1, \hat{z}_{im} \in \{0,1\}, \sum_{m=1}^{K_i} \hat{z}_{im} = 1, v_j \in \{0,1\}. \tag{13}$$

Here, when a worker is identified as adversarial, we allocate equal weight ($\frac{w_j}{K_i-1}$) to all the answers except the one the worker has chosen.[13] A locally optimal algorithm, maximizing the objective ψ_{bp} starting from an initial weight vector $\boldsymbol{w} = \epsilon\underline{\boldsymbol{1}}$, consists of the following three iterated steps:

Step 1. For fixed values of $\boldsymbol{w}$ and $\boldsymbol{v}$ and for each task i, the optimal $\hat{\boldsymbol{z}}_i$, maximizing (13), is chosen as:

$$\hat{z}_{im} = \begin{cases} 1 & \text{if } \sum_{j:r_{ij}=m} v_j w_j + \frac{1}{K_i-1}\sum_{j:r_{ij}\neq m}(1-v_j)w_j \\ & > \sum_{j:r_{ij}=k} v_j w_j + \frac{1}{K_i-1}\sum_{j:r_{ij}\neq k}(1-v_j)w_j \forall k \neq m \\ 0 & \text{otherwise} \end{cases}.$$

Step 2. For fixed values of $\hat{\boldsymbol{Z}}$ and $\boldsymbol{v}$, the optimal $\boldsymbol{w}$, maximizing (13), is given by:

$$w_k^* = \frac{\sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}v_k\delta(r_{ik}-m) + \frac{(1-v_k)}{K_i-1}(1-\delta(r_{ik}-m))}{\sqrt{\sum_{j=1}^{N}\left(\sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}v_j\delta(r_{ij}-m) + \frac{(1-v_j)}{K_i-1}(1-\delta(r_{ij}-m))\right)^2}}.$$

Step 3. For fixed values of $\underline{\hat{\boldsymbol{Z}}}$ and $\boldsymbol{w}$, the optimal $\boldsymbol{v}$, maximizing (13), is:

$$v_j = \begin{cases} 1 & \text{if } \sum_{i=1}^{T}\sum_{m=1}^{K_i} \hat{z}_{im}w_j\delta(r_{ij}-m) \geq \\ & \frac{1}{K_i-1}\hat{z}_{im}w_j(1-\delta(r_{ij}-m))\forall j \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

13. The weights across incorrect answers are normalized by $\frac{1}{K_i-1}$ so that each worker's overall contribution to task i equals his weight w_j.

3.2.2 Negative Weights Signify Adversaries

In the previous algorithm, we can see that

i) An honest worker's weight contributes to the objective function only if he votes with the (weighted) plurality.
ii) Binary parameters included to represent worker intent result in an additional optimization step. As will be seen in the experiments section, this additional step may be a source of (poor) local optima.
iii) The weights computed by step 2 will always be nonnegative, and so the full (real line) range of w_j is not being utilized.

Here, we propose an approach which remedies all three of these "issues". First, let us note that negative weights can be used to signify adversarial workers, and treated accordingly. Thus rather than apportion $\frac{w_j}{K_i-1}$ when $v_j = 0$, as in Section 3.2.1, we can equivalently apportion $\frac{|w_j|}{K_i-1}$ when $w_j < 0$ (and, as we shall see shortly, thus avoid the need to explicitly introduce binary intention parameters). Second, to appreciate a possible alternative to i), suppose an influential *nonadversarial* worker (w_j large and positive) does not agree with the (current) weighted plurality decision for a given task. Rather than *not* contributing to the plurality score and the objective function, this worker could *subtract* his weight from the scores of all answers with which he disagrees to try to *alter* the weighted plurality decision. In Appendix A, we derive an objective function that is consistent with these worker reward/penalty mechanisms. The resulting crowd aggregation problem thus becomes:

$$\max_{\hat{\boldsymbol{Z}},\boldsymbol{w}} \psi_{\text{neg}}(\boldsymbol{w},\hat{\boldsymbol{Z}}) = \sum_{j=1}^{N} w_j \Bigg[\sum_{i=1}^{T}\sum_{m=1}^{K_i} \Bigg(\hat{z}_{im}\delta(r_{ij}-m) - \frac{1}{K_i-1}\hat{z}_{im}(1-\delta(r_{ij}-m)) \Bigg) \Bigg], \tag{15}$$

$$subject\ to \sum_{j=1}^{N} w_j^2 = 1, \hat{z}_{im} \in \{0,1\}, \sum_{m=1}^{K_i} \hat{z}_{im} = 1.$$

Note that (15) accounts for both nonadversarial and adversarial workers in precisely the way we intend. If $w_j > 0$ (nonadversarial) and the worker agrees with the plurality, his full weight is added to the plurality score; if he disagrees with the plurality, $\frac{w_j}{K_i-1}$ is subtracted from all the answers with which he disagrees. If $w_j < 0$ (adversarial) and if the worker agrees with the plurality, his full weight magnitude is subtracted from the plurality score; if his answer disagrees with the plurality, his weight magnitude is equally apportioned amongst the remaining answers. We further note that, supposing $\hat{\boldsymbol{Z}}$ are the ground truth answers, then the per worker term bracketed in (15) $\sum_{i=1}^{T}\sum_{m=1}^{K_i}(\hat{z}_{im}\delta(r_{ij}-m) - \frac{1}{K_i-1}\hat{z}_{im}(1-\delta(r_{ij}-m)))$ for a spammer j goes to 0 as $T \to \infty$. This follows from the weak law of large numbers and from our assumption that a spammer will randomly choose an answer with a uniform distribution on all possible choices. Thus, spammers will (asymptotic in T) be assigned zero weights. Accordingly, (15) properly accounts for honest workers, malicious workers, and spammers. Our locally optimal algorithm for (15) consists of iteration of the following two steps, starting from the weight vector initialization:

Step 1. For a fixed $\boldsymbol{w}$ and for each task i, choose $\hat{z}_i$ as

$$\hat{z}_{im} = \begin{cases} 1 & \text{if } \sum\limits_{j:r_{ij}=m} w_j - \frac{1}{K_i-1}\sum\limits_{j:r_{ij}\neq m} w_j > \\ & \quad \sum\limits_{j:r_{ij}=k} w_j - \frac{1}{K_i-1}\sum\limits_{j:r_{ij}\neq k} w_j \ \forall\, k \neq m \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Step 2. For fixed $\hat{\boldsymbol{Z}}$, the optimal $\boldsymbol{w}$ is:

$$w_k^* = \frac{\sum\limits_{i=1}^{T}\sum\limits_{m=1}^{K_i}\hat{z}_{im}\delta(r_{ik}-m) - \frac{1}{K_i-1}(1-\delta(r_{ik}-m))}{\sqrt{\sum\limits_{j=1}^{N}\left(\sum\limits_{i=1}^{T}\sum\limits_{m=1}^{K_i}\hat{z}_{im}\delta(r_{ij}-m) - \frac{1}{K_i-1}(1-\delta(r_{ij}-m))\right)^2}}.$$

Convergence of this algorithm is ensured, again based on the proof given in Appendix A.

3.3 Semi-Supervised Case

Note that for both the methods in Sections 3.2.1 and 3.2.2, it is quite straightforward to incorporate probe task supervision. This is achieved by slightly modifying step 1 in both cases. Specifically, when a task is probe, we simply fix the value of $\hat{z}_i$ to the ground-truth value, rather than using the weighted plurality decision rule.

4 Experiments

Experiments were performed using synthetic data as well as data generated by a crowdsourced multicategory labeling task on Amazon MTurk. Additionally, for a number of UC Irvine domains, we generated a collection of heterogeneous classifiers to be used as a "simulated" crowd. We generated adversaries based on the model (2) for all our experiments. In addition to the proposed schemes, we compared with: the Dawid-Skene method [5], which uses a confusion matrix to account for worker reliabilities; simple (multicategory) plurality voting and its semi-supervised version, which exploits the probe tasks,[14] a multicategory extension of Whitehill et al. [15],[15] the multiclass method by Karger et al. [16], which performs binary quantizations, using each possible answer as threshold. We observed empirically that multiplication of the quantized answer matrix with the projection matrix L in equation (6) of [16] degraded the performance of this method. Hence for a meaningful comparison we included a modified version of [16] in our experiments where we omitted the projection

14. This is a weighted plurality scheme, where each worker is weighted by the fraction of probe tasks that he answered correctly.

15. There is an inherent problem with this method—for the case of uninformative guessing, the worker gets the answer correct fifty percent of the time irrespective of the number of categories. To remedy this, we slightly modified the approach, replacing the sigmoid $\sigma(\alpha_i\beta_j)$ with $\sigma(\alpha_i\beta_j - \tau)$, τ chosen so that $\sigma(-\tau) = \frac{1}{K_i}$.

TABLE 1
Synthetic Data Generated Using the Stochastic Model for the High Variance Regime: Changing Task Difficulty Variance

Task Variance	Average erroneous tasks									
	PLU	SS-PLU	US-DS	US-SW	US-NEG	US-GEM	SS-GEM	US-WH	US-QNT^{P}	US-QNT
4000	25.2	20.6	24.6	22.7	21.1	14.6	15.2	19.9	62.3	32.3
2000	23.1	19.4	22.8	20.4	19.8	12.9	12.8	18.2	54.1	29.3
1000	19.1	15.2	19.1	18.1	16.1	15.4	9.8	9.6	62.1	23.2
500	10.2	5.5	7.1	3.5	3.1	1.2	1.2	5.9	72.3	18.2
250	4.9	3.8	3.8	2.2	2.2	2.0	2.0	3.5	69.4	15.3

step. Let us reiterate the different methods that we will apply and study in this section:

i) PLU: Simple plurality.
ii) SS-PLU: Semi-supervised plurality.
iii) US-DS: Unsupervised Dawid-Skene algorithm [5].
iv) US-SW: Unsupervised objective-based method that uses binary parameter for intention described in Section 3.2.1.
v) SS-SW: Semi-supervised specialization of US-SW.
vi) US-NEG: Unsupervised objective-based method without intention parameters, described in Section 3.2.2.
vii) SS-NEG: Semi-supervised specialization of US-NEG.
viii) SS-GEM: Semi-supervised GEM scheme.
ix) US-GEM: Unsupervised specialization of SS-GEM.
x) US-WH: Unsupervised multicategory extension of Whitehill et al. [15]. We performed a grid search over the initial values of the α_i and β_j parameters for this method, choosing the best initialization values with respect to the incomplete data log-likelihood.
xi) US-QNT^{P}: Unsupervised quantization method of Karger et al. [16].
xii) US-QNT: Unsupervised modified quantization method of Karger et al. [16] (omitting the projection step).

4.1 Experiments with Synthetic Data

These experiments were performed in two parts. For the first part, the synthetic data was produced according to the stochastic generation described in Sections 2.2 and 2.3. The goal here was to evaluate the GEM algorithm by comparing the estimated parameters and the estimated hidden ground truth answers with their actual values used in generation of the synthetic data. We generated a crowd of 100 workers with $d_j \sim \mathcal{N}(1, 1{,}000)$, $a_j \sim \mathcal{N}(0.3, 0.2)$; 10 percent of workers were adversarial. The tasks were generated with $\tilde{d}_i \sim \mathcal{N}(20, \sigma^2)$, where σ^2 was varied. The ground truth answer for each task was chosen randomly from $\{0, 1, \ldots, 4\}$. We observed that in this regime of high task difficulty and high variance in the generated values of (both) worker skill and task difficulty, there is a definite advantage in using the GEM based schemes (SS-GEM and US-GEM) over other schemes, as shown in Table 1. We also see in Fig. 1 the high correlation between the estimated and actual values of worker skills and task difficulties. Also noteworthy is the superior performance of US-GEM over the other unsupervised objective-based schemes such as US-SW and US-NEG. In Fig. 2 we plot the histogram of worker accuracies for one trial for the first row in Table 1. This illustrates that highly skilled workers are scarce in these experiments. We also observed very poor performance of US-QNT^{P}. A comparison with US-QNT (which also compares poorly with others, including plurality voting) suggests this is due to the projection step (6) in [16]. In subsequent experiments we excluded US-QNT^{P}. The overall suboptimal performance of US-QNT (seen in the sequel) may be attributable to its quantization scheme. To illustrate, suppose that the set of answers from five workers for a task i, whose ground truth answer is 0, is $\{4, 2, 0, 0, 3\}$. Note that plurality voting will identify the correct answer (0). Now consider the quantization step in [16] which will output a vector $[1, 1, -1, -1, 1]$ for the query: "Is worker j's answer for ith task greater than 0?". This vector suggests that the ground truth answer for task i is greater than (not equal to) 0. Table 3 shows performance as a function of the number of workers assigned to each task. In each case, a random regular bipartite graph of workers and tasks was generated.

Next, we studied the "low variance" regime, generating worker skill and task difficulty parameters by $d_j \sim \mathcal{N}(1, 1)$

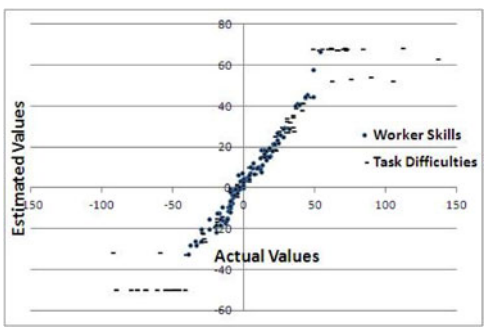


Fig. 1. Comparison of actual and estimated parameters.

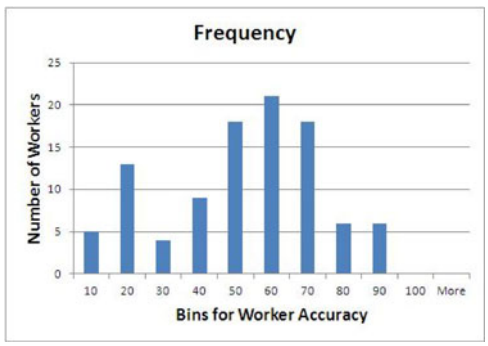


Fig. 2. Histogram of worker accuracies for high variance regime.

TABLE 2
Synthetic Data Generated Using the Stochastic Model for the Low Variance Regime: Changing Proportion of Spammers

Percentage Spammers	Average erroneous tasks									
	PLU	SS-PLU	US-DS	US-SW	US-NEG	US-GEM	SS-GEM	US-WH	US-QNT^P	US-QNT
20	17.1	16.2	14.3	13.8	12.5	16.2	15.9	16.8	28.2	61.9
30	19.1	15.9	16.3	15.2	12.7	22.9	20.3	17.3	33.4	65.3
40	32.1	29.5	30.2	29.6	25.2	33.2	32.5	27.6	41.2	66.2
50	38.1	33.6	37.2	31.2	28.3	34.8	33.6	31.9	45.6	65.8

TABLE 3
Synthetic Data Generated Using the Stochastic Model for the High Variance Regime: Changing Number of Worker Assignments

Assignment degree	Average erroneous tasks									
	PLU	SS-PLU	US-DS	US-SW	US-NEG	US-GEM	SS-GEM	US-WH	US-QNT^P	US-QNT
20	32.6	29.8	30.1	31.2	29.1	27.5	26.1	30.2	40.1	78.5
40	23.3	20.3	22.1	20.8	19.2	18.9	18.0	20.3	32.2	65.1
60	19.7	18.3	19.6	18.5	18.5	15.5	15.1	18.5	22.3	62.5
80	15.9	15.1	14.2	14.2	13.5	11.2	10.9	14.8	21.2	65.6

and $\tilde{d}_i \sim \mathcal{N}(7,1)$. We also introduced spammers (who randomly chose answers for all tasks with a uniform pmf over the answer space), whose proportion was varied. As seen in Table 2, the deterministic objective-based schemes are more suitable than GEM under this scenario. Fig. 3 shows the histogram of worker accuracies. We can observe a clear difference between worker accuracy profiles in Figs. 2 and 3. These results are illustrative of what we have observed more generally(as reported in the sequel): the GEM-based approach works best when a small minority of highly skilled experts are present amongst mostly low skilled workers with varying skill levels, whereas the objective-based approach tends to work best when a few skilled workers (not necessarily *very* highly skilled) are present in a crowd containing many spammers and the variance in skill levels across the workers is low.

In the second part we separately evaluated the deterministic objective-based schemes (US-SW and US-NEG) and compared them with simple plurality voting and US-DS. We performed this for two different models of synthetically generated data. For the first model, all workers belonged to one of three categories: spammers, "hammers" and adversarial workers. For all tasks, the spammers answered randomly, the hammers were always correct, whereas the adversarial workers always chose randomly from the incorrect answers. Table 4 shows comparison across varying proportions of spammers and adversarial workers in the crowd. We can observe that US-DS outperforms PLU, but US-SW and US-NEG give clearly superior performance. For the second model, we generated the data according to the stochastic model in Section 2.2 with the task difficulty for each task $\tilde{d}_i \sim \mathcal{U}[0,8]$. We created three categories of workers: high-skilled honest workers ($d_j \sim \mathcal{U}[0,8]$), low-skilled honest workers ($d_j \sim \mathcal{U}[0,2]$), and high-skilled simple adversarial workers ($d_j \sim \mathcal{U}[0,8]$). The adversarial workers answered incorrectly to the best of their skill level according to (2). Table 5 shows the comparison of the schemes for this model of synthetic data. Note that for both the experiments in Tables 4 and 5, we averaged the number of erroneous tasks over 20 trials. We can observe from Tables 4 and 5 that US-NEG clearly outperforms US-SW and PLU. One hypothesis for the performance advantage of US-NEG over US-SW is the additional "layer" of optimization needed to choose the binary intention parameters in US-SW. This could give a greater tendency of US-SW to find (relatively) poor local optimum solutions of its objective function. To test this hypothesis, we devised a hybrid method (US-HYB) which maximizes the US-SW objective function, but starting from $\hat{Z}$ and $\boldsymbol{v}$ initialized based on the US-NEG solution (with the sign of w_j used to initialize v_j). We can see from Table 4 that US-HYB certainly performs better than US-SW. In a separate experiment (omitted here due to space limitations), we confirmed that there is strong correlation between US-HYB achieving a greater US-SW objective function value and US-HYB achieving fewer decision errors than US-SW.

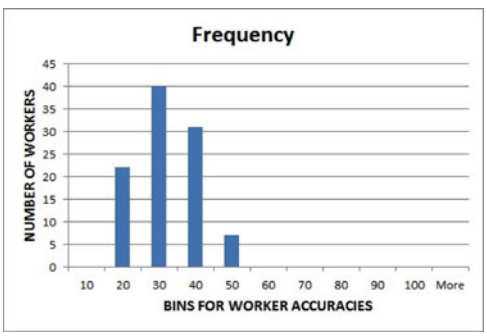


Fig. 3. Histogram of worker accuracies for low variance regime.

TABLE 4
Synthetic Data: Comparing Objective-Based Methods for Spammer, Hammer and Adversary Model

Percentage adversarial workers	Number of erroneous tasks US-SW/US-NEG/US-HYB/PLU/US-DS					
	Percentage spammers					
	10	20	30	40	50	60
0	5.4/5.1/ 5.2/ 6.9/6.2	6.7/5.9/ 6/ 8.7/8.2	6.0/5.7/ 5.6/ 9.6/8.8	10.5/9.05/ 9.1/ 16.2/13.1	10.9/9.5/ 9.7/ 16.6/13.3	13.7/11.7/ 11.4/ 22.3/17.2
5	5.6/5.4/ 5.3/ 8.1/7.6	6.6/6.3/ 6.4/ 9.8/8.2	8.7/8.6/ 8/ 13.1/11.6	10.1/9.3/ 9.4/ 16.9/14.3	12.8/11.9/ 12.0/ 20.8/18.2	19.2/17.1/ 16.9/ 29.5/24.2
10	6.8/6.1/ 6.0/ 10.4/9.1	7.3/6.2/ 6.6/ 11.9/10.3	8.8/8.0/ 8.5/ 15.8/12.2	10.3/9.4/ 9.5/ 18.3/16.6	13.8/12.4/ 12.3/ 24/19.4	29.7/16.2/ 16.3/ 30.7/25.3
15	5.8/5.0/ 5.0/ 9.7/9.1	8.4/7.5/ 7.7/ 14.4/12.3	10.9/9.8/ 9.9/ 18.1/16.2	13.5/12.5/ 12.7/ 23.4/20.5	16.6/14.8/ 15.5/ 30.2/25.9	24.3/21.0/ 21.3/ 37.8/31.2
20	8.1/7.1/ 7.5/ 13.8/12.3	8.3/3.7/ 8.5/ 16.6/14.2	12.2/11.2/ 11.3/ 22.2/18.6	13.7/11.8/ 12.2/ 27/23.6	18.8/16.8/ 16.7/ 34.2/30.1	29.2/22.3/ 22.4/ 44.8/37.9

TABLE 5
Synthetic Data: Comparing Objective-Based Methods for Data Generated Using Model in Section 2.2

Percentage high skilled adversarial workers	Number of erroneous tasks PLU/US-SW/US-NEG/US-DS					
	Percentage low skilled honest workers					
	10	20	30	40	50	60
5	9.4/6.7/6.0/9.6	10.5/6.5/6.1/ 10.4	12.7/6.9/6.7/ 12.5	14.4/8.3/7.2/14.2	20.3/11.3/10.5/ 19.4	26.9/17.3/ 15.4/23.6
10	9.8/6.9/6.1/9.7	12.1/6.4/6.1/ 12.5	15.8/8.6/7.7/ 15.9	18.4/10.5/9.8/19.1	23.1/13.2/12.6/ 23.2	29.5/18.3/16/ 29.8
15	10.5/6.0/5.6/ 10.2	13.5/8.2/7.1/ 13.3	16.7/9.4/7.7/ 16.7	22.9/12.8/11.3/23	27.7/14.5/13.7/ 27.6	37/22.4/20.1/ 36.9
20	12.8/6.7/6.4/ 12.5	17.4/9.7/8.5/ 16.9	20.4/9.1/8.4/ 20.1	25.7/14.7/12.6/24.7	37.4/20.3/18.2/ 35.2	42.5/27.4/ 22.4/38.1

4.2 Simulating a Crowd Using an Ensemble of Classifiers

We also leveraged ensemble classification to generate a set of automated workers (each an ensemble classifier) using boosting [17]. Each such classifier (worker) is a strong learner obtained by applying multiclass boosting to boost decision tree-based weak learners. The strength (accuracy) of each worker was varied by controlling the number of boosting stages. Each weak learner was trained using a random subset of the training data to add more heterogeneity across the workers' hypotheses. Note that unlike Section 4.1, this approach to simulated generation of a crowd is *not* obviously matched to our stochastic data generation model in Section 2.2. Thus, this more complex simulation setting provides a more realistic challenge for our model and learning. We ran Multiboost [18] 100 times to create a crowd of 100 workers for four domains that are realistic as crowdsourcing domains: Pen Digits,[16] Vowel, Dermatology, and Nominal.[17] For each experimental trial, 100 crowd tasks were created by randomly choosing 100 data samples from a given domain; 10 of them were randomly chosen to be probe. The rest were used for training the strong (ensemble) classifiers/ workers. The average of the number of crowd-aggregated erroneous tasks was computed across five trials, where each trial consisted of a freshly generated crowd of workers and set of tasks. In Tables 6, 7, 8, and 9 we give performance for different worker accuracy means and variances for the four domains. We did not directly control these values since they were an outcome of the boosting mechanism.

However we could control the number of boosting stages used by each worker. We also show the performance when 10 percent of the workers were replaced by adversarial workers. These synthetic adversaries retained the estimated skill level of the (replaced) workers and generated their answers using the stochastic model described in Section 2.3. For these tables, worker accuracy mean and variance are estimated excluding the adversarial workers.

In Table 6, for Pen Digits, we can see the gain in performance for our methods, especially in the presence of adversarial workers. Note that for low means of worker accuracy, the GEM based methods outperform others, whereas the weighted plurality based methods using negative weights for adversaries (US-NEG and SS-NEG) performed better than other schemes for relatively higher means of worker accuracy. Figs. 4 and 5 show the histogram of worker accuracies for low and high mean cases, respectively. Fig. 4 corresponds to the first row in Table 6 (for a single trial). Here we see a highly skewed distribution of worker accuracies where a tiny minority of high-skilled workers exists among

16. We resampled the data set to have only odd digits.

17. Hungarian named entity data set [19]. Identifying and classifying proper nouns into four categories: not a proper noun, person, place, and organization.

TABLE 6
Experiments Using Pen Digits Data Set: Average Number of Erroneous Tasks

Worker accuracy mean	Worker accuracy variance	Task accuracy variance	Without adversarial workers With 10% adversarial workers										
			PLU	SS-PLU	US-DS	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
24.62	289.2	28.42	22.1	11.1	27.45	12.2	11.9	10.2	10.2	8.6	8.4	21.7	32.8
			43.6	13.8	46.2	14.1	13.8	11.5	11.3	5.6	5.1	40.6	38.9
27.82	323.2	39.9	20.2	8.9	19.3	11.5	11.2	10.7	10.4	8.1	8.1	17.1	31.4
			26.4	14.3	26.1	14.1	13.9	12.6	12	6.2	5.6	24.5	38.9
33.6	490.6	96.3	14.1	9.6	13.8	8.6	8.6	6.4	6.4	8.8	8.4	12.6	22.4
			18.9	9.6	17.6	10.1	10.1	6.8	6.7	8.1	7.5	17.9	30.6
40.3	581.2	204.4	11.8	6.5	11.4	6.6	6.6	5.2	5.1	7.8	7.5	10.9	18.6
			14.3	7.8	13.5	7.8	7.8	4.9	4.9	7.1	6.9	13.8	22.5

TABLE 7
Experiments Using Dermatology Data Set: Average Number of Erroneous Tasks

Worker accuracy mean	Worker accuracy variance	Task accuracy variance	Without adversarial workers With 10% adversarial workers										
			PLU	SS-PLU	US-DS	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
18.8	221.5.2	46.4	48.2	24.2	41.5	25.6	23.9	16.8	15.4	24.6	10.5	41.5	50.1
			63.4	21.6	63.8	40.3	38.7	43.1	37.7	35.7	8.4	52.6	62.8
23.7	441.8	85.4	35.8	14.7	32.7	13.2	13.1	6.5	6.4	15.2	7.1	28.6	40.2
			46.5	11.5	45.2	25.3	23.9	18.4	18.2	13.8	5.8	35.9	45.6
29.5	654.2	92.3	28.2	9.4	25.1	11.4	11.2	5.9	5.9	4.8	4.1	25.4	35.6
			33.1	10.6	31.9	9.8	9.5	10.2	10.1	2.9	2.1	32.6	42.6
34.2	689.1	145.2	11.1	3.8	11	3.8	3.7	3.8	3.8	3.4	2.9	9.8	20.1
			20.6	6.4	17.6	3.5	3.5	3.2	3.1	3.3	2.5	15.6	28.6

TABLE 8
Experiments Using Vowel Data Set: Average Number of Erroneous Tasks

Worker accuracy mean	Worker accuracy variance	Task accuracy variance	Without adversarial workers With 10% adversarial workers										
			PLU	SS-PLU	US-DS	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
21.3	352.6	34.5	38.4	17.2	32.3	26.4	26.4	22.8	22.8	22.6	21.3	31.6	49.8
			45.6	13.8	34.8	25.4	25.3	20.9	19.9	14.2	13.8	32.8	43.6
26.4	420.3	52.8	25.5	15.3	22.3	15.8	15.8	14.2	14.2	18.1	17.8	24.5	30.6
			28.2	13.9	25.6	16.5	16.3	12.6	11.6	12.1	10.1	25.4	34.5
30.9	567.2	65.2	18.6	13.2	18.2	12.8	12.8	11.5	11.2	14.1	13.6	17.1	25.6
			26.7	15.1	20.3	12.6	12.3	13.2	12.7	8.6	7.9	22.6	35.8
35.4	708.9.2	98.2	16.3	13.9	16.2	14.4	14.4	14.4	14.1	17.8	16.2	14.4	25.2
			24.2	14.8	24.4	13.5	13.5	12.9	12.2	13.6	10.3	22.6	28.9

TABLE 9
Experiments Using Nominal Data Set: Average Number of Erroneous Tasks

Worker accuracy mean	Worker accuracy variance	Task accuracy variance	Without adversarial workers With 10% adversarial workers										
			PLU	SS-PLU	US-DS	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
43.8	543.8.3	356.7	21.2	17.6	20.8	19.6	19.6	19.6	19.6	19.5	19.5	19.8	28.6
			24.3	19.3	23.2	19.8	19.8	19.2	18.8	18.2	17.5	21.6	31.7
48.6	623.6	324.7	10.3	8.2	10.3	9.6	9.4	8.8	8.8	7.3	6.2	9.8	15.6
			11.9	8.3	11.8	9.2	9.1	7.4	7.3	6.8	5.8	11.2	18.6
50.2	648.3	328.4	9.8	7.5	9.7	7.8	7.8	7.8	7.7	7.6	5.5	9.7	15.8
			12.5	6.3	12.8	8.2	8.2	7.2	7.2	6.4	5.3	11.6	15.9
53.8	690.3	267.8	9.3	5.8	9.3	7.2	7.2	5.9	5.9	5.1	4.2	8.8	14.6
			10.5	4.8	10.3	6.8	6.8	5.1	5	3.4	3.1	10.2	15.6

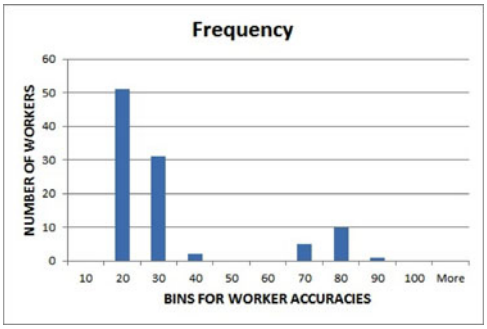


Fig. 4. Pen Digits data set: Histogram of worker accuracies with a skewed distribution of skills.

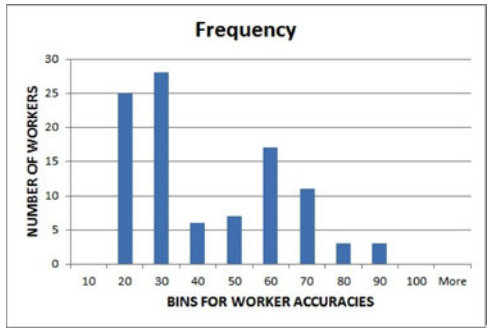


Fig. 5. Pen Digits data set: Histogram of worker accuracies with a more gradual "spread" of skill distribution.

spammers or very low-skilled workers. As already known from our experiments with synthetic data, this distribution seems to best "agree" with the GEM based schemes. The histogram in Fig. 5 corresponds to the last row in Table 6. Here we observe a less skewed distribution of worker accuracies. The objective based US-NEG and SS-NEG schemes perform best in this regime. The GEM based scheme seems more robust to adversaries than other methods, achieving improved crowd aggregation accuracy, relative to where no adversaries are present, for almost all the entries in Table 6. Also observed from this table is that 10 percent probe task supervision does not greatly improve performance, e.g., US-GEM performs as well as SS-GEM. Moreover, SS-PLU is highly sub-optimal in comparison with the unsupervised methods US-GEM and US-NEG. From these results, we can see that our proposed schemes are able to identify and leverage the expertise of a small subset of highly skilled workers, thus defeating "tyranny of the masses". An interesting observation from the results on Dermatology in Table 7 is that although unsupervised, US-NEG performs very close to SS-GEM when adversarial workers are absent. Overall on this data set, SS-GEM greatly outperforms other methods. Also, unlike Pen Digits, probe supervision greatly assists the inference using GEM for lower mean worker accuracy (as seen in the first two rows of Table 7). Table 8 shows the results on the vowel data set. Here, US-NEG and SS-NEG clearly outperform all other methods, even SS-GEM, in the absence of adversaries. US-GEM and SS-GEM, perform better than all others when adversarial workers are introduced. We observed from our experiments with the Nominal data set (Table 9) that the GEM based methods perform better than others overall, although the performance gap diminishes as the mean worker accuracy improves.

4.3 Experiments with Heterogeneity of Answer Cardinality

We considered UCI Pen Digits and Dermatology as in Section 4.2, but randomly excluded some incorrect answers from a given task, thus creating a "condensed" task-dependent answer set. The cardinality of this set for a given task was between 2 to 5 for Pen Digits and between 2 and 6 for Dermatology. Workers whose incorrect answers for a given task were not in the condensed set had their answer randomly reassigned to an incorrect answer that is in the condensed set. Tables 10 and 11 show the comparison of error rates for different methods. Note that we omitted Dawid-Skene [5] since it is based on a full confusion matrix. We can observe that SS-GEM greatly outperformed others in the absence of adversarial workers, whereas, both US-GEM and SS-GEM performed substantially better in comparison to other methods in the case of 10 percent adversarial workers.

4.4 MTurk Experiment

We designed an image labeling task where workers had to provide the country of origin, choosing from *Afghanistan, India, Iran, Pakistan, Tajikistan* (Fig. 6). Some of the regions in these countries are very similar in their culture, geography, and demography and hence only people with domain experience and a deep understanding of the region will likely know the true answer. For instance, the blue rickshaws are typical to Pakistan and the yellow taxis are more common in Kabul. One can also guess, e.g., from the car models on the street or from the script on street banners. We posted a task consisting of 50 such images on Amazon MTurk and asked all workers to upload a file with their answers on all tasks.

We received responses from 62 workers. In order to evaluate under the scenario where workers answer only a subset of the tasks, for each task, we used answers from a sampled set of workers using a randomly generated degree regular bipartite graph consisting of worker and task nodes. A worker's answer to a task was used only when a link existed in the bipartite graph. Table 12 shows the average number of erroneous crowd-aggregated answers as we varied the number of tasks assigned to each worker. The average was computed over five trials, each involving a new graph instance and using five

Fig. 6. The MTurk experiment: a few of the sample images.

TABLE 10
Experiments Using Pen Digits Data Set: Heterogeneous Answer Space Cardinality

Worker accuracy mean	Worker accuracy variance	Task accuracy variance	Without adversarial workers With 10% adversarial workers									
			PLU	SS-PLU	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
33.05	443.5	60.3	38.2	19.1	27.6	27.4	26.1	26.1	37.5	11.2	37.9	44.2
			45.8	20.6	28.2	28.2	20.2	20.1	10.7	9.1	34.2	47.8
39.77	383.4	197.9	41.2	22.6	21.2	23.0	19.4	19.2	33.4	7.8	38.3	45.7
			46.4	19.3	22.5	21.9	16.9	16.9	6.3	6.1	35.8	50.2
44.19	352.4	314.2	35.3	22.0	22.2	22.2	22.2	22.0	28.6	6.9	42.3	49.2
			44.1	18.8	23.6	23.6	15.5	15.4	6.9	6.5	46.1	50.5
47.32	427.5	3342.7	30.2	20.9	23.5	23.2	20.6	20.6	25.3	7.2	27.8	37.2
			37.1	17.6	22.9	22.9	19.8	19.8	6.8	6.6	32.4	42.6

TABLE 11
Experiments Using Dermatology Data Set: Heterogeneous Answer Space Cardinality

Worker accuracy mean	Worker accuracy variance	Task accuracy variance	Without adversarial workers With 10% adversarial workers									
			PLU	SS-PLU	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
29.63	789.7	65.4	42.6	9.6	8.5	8.5	8.4	8.4	40.6	4.1	35.2	50.2
			44.8	8.2	8.2	8.2	7.6	7.6	4.8	4.1	25.6	52.8
34.74	276.79	98.69	35.4	7.8	5.6	5.6	5.8	5.8	33.2	3.2	33.3	44.7
			37.1	6.3	3.8	6.7	4.5	4.5	3.8	3.2	31.8	45.2
39.24	651.3	223.4	31.2	8.2	6.5	6.5	5.3	5.3	27.6	2.6	30.4	35.6
			33.3	7.6	5.8	5.8	4.1	4	3.5	2.2	30.6	38.2
46.45	1104.2	240.6	16.2	6.5	5.4	5.3	3.8	3.6 4	13.8	1.5	13.2	21.3
			17.3	6.2	5.2	5.2	4.2		2.8	2.6	15.4	22.6

TABLE 12
MTurk Experiment: Average Number of Erroneous Tasks

Assign-ment Degree	PLU	SS-PLU	US-DS	US-SW	SS-SW	US-NEG	SS-NEG	US-GEM	SS-GEM	US-WH	US-QNT
10	15.6	18.1	15.4	14.9	14.9	14.3	14.3	13.8	13.4	15.6	25.2
20	14.6	13.3	14.7	14.1	14.1	13.6	13.7	7.4	7.4	14.3	26.5
30	14.2	10.9	14.3	13.2	13.2	13	12.9	5.1	5.1	14.1	23.2
40	12.3	7.7	13	11.3	11.2	10.4	10.3	4.4	4.2	11.7	20.2
50	13.6	8	13.2	13.4	12.1	11.5	11.5	3.4	3.4	11.9	19.1

randomly chosen probe tasks. The histogram of worker accuracies is shown in Fig. 7. From the histogram we can observe that only a small fraction of workers could answer most tasks correctly.

4.5 Summary of Results

An overarching observation is the big performance gain of our methods (both GEM and objective-based) over the baseline methods when the average task difficulty far exceeds the average worker skill level, resulting in a minority of workers competent enough to solve most of the tasks.

In this scenario, our results suggest that high variance in worker skill and task difficulty is more favorable for the GEM based methods, whereas the objective based methods perform better for low variance of worker skills and task difficulty. This can be attributed to the fact that the GEM based methods account for the task difficulty whereas the objective-based methods do not. We also observed that as the gap between average worker skill and average task difficulty narrows, the performance gap between the proposed methods and the baseline methods decreases. For the MTurk experiment, we chose a task (geographic and demographic discretion) that introduced both high average task difficulty and variation in task difficulty (for instance, ranging from the picture of US marines in Afghanistan to the mountains in northern India that look more Central Asian than South Asian). As observed from our synthetic and simulated worker experiments, these conditions are best addressed by the GEM based schemes. Profiling the

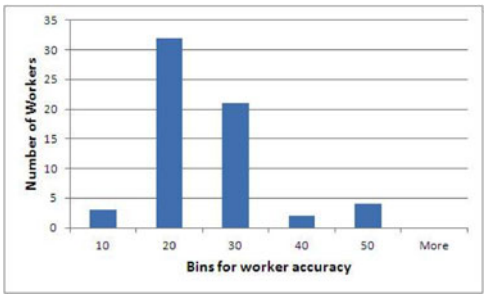


Fig. 7. Histogram of worker accuracies for the MTurk experiment.

computational performance of both the approaches suggests that the objective-based methods are faster than GEM based with a more deterministic turnaround time. On a system with 2.5 GHz dual-core Intel Core i5 processor with 3MB L3 cache and 4 GB of $1{,}600$ MHz DDR3 memory, US-NEG took an average time of 0.623 seconds with a variance of 0.027 seconds across five trials corresponding to the first row in Table 11. Whereas for the same experiment, US-GEM took an average of 5.28 seconds with a variance of 2.64 seconds.

5 Related Work

Stochastic models for generation of workers' answers have been previously considered in [1], [5], [12], [20]. A related multiple label learning problem is considered in [21]. In the Dawid-Skene model [5], the parameters are learned via an EM algorithm. Their approach generalizes our representation of a worker's skill by using a per-worker confusion matrix, but requires learning more parameters per worker while assuming equal difficulty across tasks. Note that this method's performance was relatively poor in our experiments when the variance of task difficulty was high. A similar approach was used in [22] to label volcanoes in satellite images of Venus. Some models such as [21] and [20] also include and exploit additional (domain-specific) features derived from the tasks. Our work does not exploit such information. Modeling task difficulty has been considered previously in [12] and [1] for the binary case. A multicategory extension of [12] was proposed by the authors in [23]. However this approach does not plausibly model a worker's response in a K-category ($K > 2$ category) setting. Specifically, in [23], an honest worker answers an impossible task (a task with infinite difficulty parameter) with a probability equal to $1/2$, instead of $1/K$. A model incorporating variable task difficulty in a multicategory setting was considered in [24]. However, [24] does not estimate individual task difficulty and their minimax entropy-based approach necessitates regularization using multiple hyper-parameters, which need to be chosen via methods such as cross-validation. In our model, we do not require any hyper-parameters. Moreover, unlike their work, we explicitly model interplay between worker skill and task difficulty.

Relative to this body of work, the primary novel contribution of our GEM-based approach lies in jointly considering and modeling worker skill, task difficulty and adversarial worker behavior in a multicategory setting that also accommodates task-dependent answer spaces. Our representation of worker skill and task difficulty in a common space, and our associated parametric model for producing correct answers is also novel. To the best of our knowledge, our objective-based approaches have also not been previously proposed in the crowdsourcing (and more general decision aggregation) literature.

Our focus has been mostly on tasks that require an expertise (skill) present only in a minority of the crowd. In [12] and [1], task difficulty was considered explicitly, but only for the binary case. Relative to these works, our model's use of task difficulty (compared against worker skill) is novel. Unlike our work, [1] assumes all tasks are drawn from the same (classification) domain.

Adversarial workers in the binary case were accounted for in [12], [25], and [6]. Here, we have characterized adversarial behavior for a more general (multicategory) setting and proposed a realistic model for more sophisticated adversaries that generalizes the random "spammer". Compared with [26] and [8], our characterization of adversarial behavior also accounts for task difficulty. Particularly, our method exploits adversarial workers on the low and moderate difficulty tasks, for which they almost always answer incorrectly. These dynamics are inherent to our model described in Section 2.2. We also showed how we can retain the interpretation of negative weights as indicating adversaries, in generalizing from the binary [6] to the multicategory case. The authors of [6] and [27] consider other statistical methods for crowd aggregation, such as correlation-based rules and low rank matrix approximation. These methods have been studied for binary classification tasks. Our objective-based approach generalizes the weighted majority theme of these papers to a multicategory case, incorporating honest workers, adversaries, and spammers. We note that, recently, [16] extends the low rank approximation approach to the multicategory case in a homogeneous task difficulty setting using multiple binary mappings (quantizations) of the multicategory answer space (with the answer index either greater than or less than a set of thresholds). In our experiments, this approach was not found to give competitive performance.

6 Extensions and Future Work

Our adversarial model accounts for workers that are malicious, but not in a very crafty way. Here, we present a model for more strategic adversaries. Such a worker answers *correctly* for the simpler tasks, with difficulty below a certain level, in an attempt to evade detection. Assume $\theta_j < d_j$ to be such a threshold for worker j. The pmf β for this (complex) adversarial worker is given by:

$$\beta(r_{ij} = l|\Lambda_{ij}, v_j = 0, z_i) = \begin{cases} \left(\frac{1}{K_i - 1}\right)\left(\frac{1}{1+e^{-b_j(\theta_j - \tilde{d}_i)}}\right)\left(\frac{1}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \\ \quad + \left(\frac{1}{K_i}\right)\left(\frac{e^{-a_j(d_j - \tilde{d}_i)}}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \text{if } l = z_i \\ \left(\frac{1}{K_i - 1}\right)\left(\frac{e^{-b_j(\theta_j - \tilde{d}_i)}}{1+e^{-b_j(\theta_j - \tilde{d}_i)}}\right)\left(\frac{1}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \\ \quad + \left(\frac{1}{K_i}\right)\left(\frac{e^{-a_j(d_j - \tilde{d}_i)}}{1+e^{-a_j(d_j - \tilde{d}_i)}}\right) & \text{otherwise.} \end{cases} \quad (17)$$

Here, essentially, the worker answers correctly with high probability for easy tasks ($\theta_j > \tilde{d}_i$), he excludes the correct answer for more difficult tasks below his skill level, and for even more difficult tasks that defeat his skill level ($d_j < \tilde{d}_i$), he answers correctly at random ($\frac{1}{K_i}$).

In the future, we would like to comprehensively evaluate more complex adversarial models, including the one proposed above. We would also like to explore collusion attacks, where a group of adversarial workers collude and submit the same (but incorrect) answer for a task. This may severely affect the performance of unsupervised methods. We can also consider the case of heterogeneous tasks where there *is* domain annotation. In this case, a worker can have separate skill parameters for each domain present in the batch. Our methods guarantee only locally optimal solutions. However, we observed that an unbiased initialization of parameters, i.e., setting the same initial value of parameters across all workers and tasks, robustly yields accurate parameter estimates in the case of synthetic data. Moreover, our approaches achieved performance superior to the comparison methods irrespective of any local optimum issues. We may, in future, also investigate approaches for avoiding local optima.

APPENDICES

A.A Derivation of the Objective Function in (15)

Assume each worker i answers each task j independently, with probability p_{ij} of getting the right answer. Given knowledge of these probabilities, the estimated answers, maximizing the data log-likelihood, solve:

$$\max_{\hat{\boldsymbol{z}}} \sum_i \sum_m \hat{z}_{im} \sum_j \Bigg(\delta(r_{ij} = m) \log(p_{ij}) + \frac{1}{K_i - 1} \delta(r_{ij} \neq m) \log(1 - p_{ij}) \Bigg). \tag{18}$$

Assuming homogeneity of difficulty across tasks, i.e., $p_{ij} = p_j \; \forall i$, we have

$$\max_{\hat{\boldsymbol{z}}} \sum_i \sum_m \hat{z}_{im} \sum_j \Bigg(\delta(r_{ij} = m) \log(p_j) + \frac{1}{K_i - 1} \delta(r_{ij} \neq m) \log(1 - p_j) \Bigg). \tag{19}$$

Using Taylor approximation around $\frac{1}{2}$ and replacing $\log(x)$ with $(2x - 1) + \log(\frac{1}{2})$, we get,

$$\max_{\hat{\boldsymbol{z}}} \sum_i \sum_m \hat{z}_{im} \sum_j \Bigg(\delta(r_{ij} = m)(2p_j - 1) + \frac{1}{K_i - 1} \delta(r_{ij} \neq m)(1 - 2p_j) \Bigg), \tag{20}$$

and letting $w_j = 2p_j - 1$, we get,

$$\max_{\hat{\boldsymbol{z}}} \sum_i \sum_m \hat{z}_{im} \sum_j \Bigg(w_j \delta(r_{ij} = m) - \frac{1}{K_i - 1} w_j \delta(r_{ij} \neq m) \Bigg).$$

B. Proof of Convergence for Theorem 3.1

The proof follows along the lines of standard proofs of convergence of clustering algorithms, e.g., [28]. In choosing an answer for each task according to (9), one finds the task "partition" that globally maximizes (8), given the workers' weights held fixed. Likewise, given the answers held fixed, (10) is a convex optimization problem with respect to the workers's weights, with a closed form solution given by (12). Thus, both steps in an iteration are non-decreasing in (8). This implies that the algorithm cannot return, in the next iteration, to a partitioning that yields a smaller objective than the current one. However, since there are a finite number of partitions, there is a finite set of possible objective function values that can be achieved by any iteration. Thus, after a finite number of iterations, e.g., L, the value of (8) at iterations $L - 1$ and L must be the same (Moreover, the solutions will be the same at these iterations, unless there is a tie between the best answers (based on (9)) for at least one task). The algorithm thus terminates in a finite number of iterations and, if there are no ties, at a fixed point. Let W_L be the vector of workers' weights at termination (at iteration L). Also, let Z_L be the binary matrix whose $1's$ indicate the crowd-aggregated answers for all tasks at iteration L. To show that this fixed point solution is also a local maximum of (8), consider all other weight vector solutions $\tilde{W}$, where $\|W_L - \tilde{W}\|^2 \leq \delta$, $\delta > 0$. Since there are a finite number of partitions, δ can be made sufficiently small, such that the optimal partition, given $\tilde{W}$, is the same, and equal to Z_L, for all $\tilde{W}$ within the δ-neighborhood of W_L. However, W_L maximizes (8) given the partition Z_L. Thus, W_L and Z_L determine a local maximum of (8).

REFERENCES

[1] P. Welinder, S. Branson, S. Belongie, and P. Perona, "The multidimensional wisdom of crowds," *Adv. Neural Inf. Process. Syst.*, vol. 6, no. 7, p. 8, 2010.

[2] M. Vukovic, "Crowdsourcing for enterprises," in *Proc. IEEE World Conf. Services-I*, 2009, pp. 686–692.

[3] J. R. Douceur, "The sybil attack," in *Proc. Revised Papers First Int. Workshop Peer-to-Peer Syst.*, 2002, pp. 251–260.

[4] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," in *Proc. ACM SIGCOMM Workshop Practice Theory Incentives Netw. Syst.*, 2004, pp. 228–236.

[5] A. Dawid, and A. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Appl. Statist.*, vol. 28, pp. 20–28, 1979.

[6] D. Karger, S. Oh, and D. Shah, "Iterative learning for reliable crowdsourcing systems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1953–1961.

[7] P. Welinder and P. Perona, "Online crowdsourcing: Rating annotators and obtaining cost-effective labels," in *Proc. IEEE Conf Comput. Vis. Pattern Recog. Workshops*, 2010, pp. 25–32.

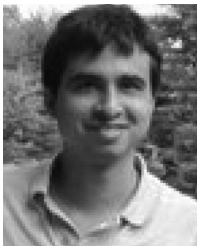
[8] V. C. Raykar, and S. Yu, "Eliminating spammers and ranking annotators for crowdsourced labeling tasks," *J. Mach. Learn. Res.*, vol. 13, pp. 491–518, 2012.

[9] X. L. Meng and D. Van Dyk, "The EM algorithman old folk-song sung to a fast new tune," *J. Roy. Statist. Soc.: Series B (Statist. Methodol.)*, vol. 59, no. 3, pp. 511–567, 1997.

[10] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statist. Soc. Series B (Methodol.)*, vol. 39, pp. 1–38, 1977.

[11] K. Bache and M. Lichman, "UCI machine learning repository," SchoolInf. Comput. Sci., Univ. California, Irvine, CA, USA, 2013.

[12] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," *Adv. Neural Inf. Process. Syst.*, vol. 22, pp. 2035–2043, 2009.
[13] M. W. Graham, and D. J. Miller, "Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1289–1303, Apr. 2006.
[14] C. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.
[15] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Supplementary derivation for whose vote should count more. (2009). [Online]. Available: http://mplab.ucsd.edu/ jake/supp.pdf
[16] D. Karger, S. Oh, and D. Shah, "Efficient crowdsourcing for multiclass labeling," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, 2013, pp. 81–92.
[17] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.
[18] D. Benbouzid, R. Busa-Fekete, N. Casagrande, F. Collin, and B. Kégl, "MULTIBOOST: A multi-purpose boosting package," *J. Mach. Learn. Res.*, vol. 13, pp. 549–553, 2012.
[19] G. Szarvas, R. Farkas, and A. Kocsor, "A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms," in *Proc. 9th Int. Conf. Discovery Sci.*, 2006, pp. 267–278.
[20] V. Raykar, S. Yu, L. Zhao, G. Valadez, C. Florin, L. Bogoni, and L. Moy, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, pp. 1297–1322, 2010.
[21] R. Jin, and Z. Ghahramani, "Learning with multiple labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2002, pp. 897–904.
[22] P. Smyth, U. Fayyad, M. Burl, P. Perona, and P. Baldi, "Inferring ground truth from subjective labelling of venus images," in *Proc. Adv. Neural Inf. Process. Syst.*, 1995, pp. 1085–1092.
[23] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Supplementary materials. (2009). [Online]. Available: http://mplab.ucsd.edu/~'jake/supp.pdf
[24] D. Zhou, J. Platt, S. Basu, and Y. Mao, "Learning from the wisdom of crowds by minimax entropy," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2204–2212.
[25] A. Ghosh, S. Kale, and P. McAfee, "Who moderates the moderators?: Crowdsourcing abuse detection in user-generated content," in *Proc. 12th ACM Conf. Electron. Commerce*, 2011, pp. 167–176.
[26] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on amazon mechanical turk," in *Proc. ACM SIGKDD Workshop Human Comput.*, 2010, pp. 64–67.
[27] D. R. Karger, S. Oh, and D. Shah, "Budget-optimal crowdsourcing using low-rank matrix approximations," in *Proc. 49th IEEE Annu. Allerton Conf. Commun., Control, Comput.*, 2011, pp. 284–291.
[28] R. M. Gray, J. C. Kieffer, and Y. Linde, "Locally optimal block quantizer design," *Inf. Control*, vol. 45, no. 2, pp. 178–198, 1980.



Aditya Kurve received the Master's and PhD degrees in electrical engineering from The Pennsylvania State University, University Park, in 2011 and 2013, respectively. He is currently working as a forensic data scientist at Pindrop Security, Atlanta, Georgia. His research interests include behavioral modeling of multi-agent systems and knowledge discovery using machine learning and game-theoretic tools. He has previously worked with MathWorks, Sasken Communication Technologies, and Conexant Systems.

David J. Miller received the BSE degree from Princeton University in 1987, the MSE degree from the University of Pennsylvania in 1990, and the PhD degree from the University of California, Santa Barbara (UCSB) in 1995, all in electrical engineering. He was also employed as a communications engineer by General Atronics Corp., Wyndmoor, Pennsylvania, from 1988 to 1990. He joined Penn State's Electrical Engineering Department in 1995, and was promoted to a tenured associate professor in 2001 and to full professor in 2007. He is an active researcher in the areas of machine learning, data compression, image processing, bioinformatics, data mining, and intrusion detection for computer networks. He received a US NSF CAREER Award in 1996. He was the chair of the Machine Learning for Signal Processing Technical Committee, within the IEEE Signal Processing Society from 2007 to 2009. He was an associate editor for *IEEE Transactions on Signal Processing* from 2004 to 2007. He was the general chair for the 2001 IEEE Workshop on Neural Networks for Signal Processing. He is a senior member of the IEEE.

George Kesidis received the MS and PhD degrees in EECS from U.C. Berkeley in 1990 and 1992, respectively. He was a professor in the E&CE Department at the University of Waterloo, Canada, from 1992 to 2000. Since 2000, he has been a professor of CSE and EE at the Pennsylvania State University. His research, including several areas of computer/communication networking and machine learning. His research has been primarily supported by the NSERC of Canada, NSF, and Cisco Systems URP. He served as the TPC co-chair of IEEE INFOCOM 2007 among other networking conferences. He has also served on the editorial boards of the *Computer Networks Journal*, *ACM TOMACS*, and the *IEEE Journal on Communications Surveys and Tutorials*. Currently, he is an "intermittent expert" (part-time program officer) for the US National Science Foundation's Secure and Trustworthy Cyberspace (SaTC) program. His home page is http://www.cse.psu.edu/~kesidis