Multicategory Crowdsourcing Accounting for Plurality in Worker Skill and Intention, Task Difficulty, and Task Heterogeneity

Aditya Kurve
Department of Electrical Engineering
The Pennsylvania State University
ack205@psu.edu

David J. Miller
Department of Electrical Engineering
The Pennsylvania State University
djmiller@engr.psu.edu

George Kesidis
Department of Electrical and CSE Engineering
The Pennsylvania State University
kesidis@engr.psu.edu

Abstract

Crowdsourcing allows one to instantly recruit workers on the web to annotate image, web page, 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 allows us to model a broad range of worker types. Our deterministic objective-based approach does not assume a model for worker response generation. Instead, it 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 semisupervised settings, and also when the batch of tasks is heterogeneous. 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.

I. INTRODUCTION

Crowdsourcing systems leverage the 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 and Crowd Flower, which provide a systematic, convenient and templated 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, writing and reviewing software code, designing products, and also raising funds. Here, we focus on crowdsourcing tasks with a categorical answer space.

Although the crowd expedites annotation, its anonymity allows noisy or even malicious labeling to occur. Online reputation systems can help reduce the effect of noisy labels, but are susceptible to Sybil or whitewashing 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

1 www.mturk.com
2 www.crowdflower.com
3 www.topcoder.com
4 www.crowdfunding.com
the ground truth answer. The estimation may use simple voting or more sophisticated aggregation methods, e.g., [3, 9, 20]. The aggregation approaches we propose in this work have several notable characteristics. First, we make a clear distinction between worker skill level and worker intention, i.e., whether the worker is honest or malicious. This dissection allows us to plausibly characterize the behavior of an adversarial worker in a multiclass 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 non malicious workers). Second, some approaches are only suitable for binary (two choices) tasks and assume that the batch of tasks to be solved all come from the same (classification) domain [20]. By contrast, our approaches explicitly address multiclass tasks and, further do not require that all tasks be drawn from the same domain, i.e., the batch of tasks can be a heterogeneous mix of problems. We propose two distinct frameworks for jointly estimating workers’ skills, intentions, and the ground truth answers for the tasks.

Our first approach assumes a stochastic model for answer generation that plausibly captures the interplay between worker skills, intentions, and task difficulties. To the best of our knowledge, this is the first model that incorporates the difference between worker skill and task difficulty (measured on the real line) in modeling the accuracy of workers on individual tasks. In this model, we formalize the notion of an adversarial worker and discuss and model different types of adversaries. A simple adversary gives incorrect answers “to the best of his skill level”. More “crafty” adversaries can attempt to evade detection by only giving incorrect answers on the more difficult tasks solvable at their skill level. The detection of adversaries and the estimation of both worker skills and task difficulties can be assisted by the knowledge of ground-truth answers for some (probe) tasks. Unlike some papers such as [15], we assume that all our probe tasks are labeled by (non-malicious) experts. Accordingly, we formulate a semisupervised approach, invoking a generalized EM (GEM) algorithm [11] 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 general approach specializes to an unsupervised method when no (labeled) probe tasks are available. Interestingly, our crowd aggregation rule comes precisely from the E-step of our GEM algorithm, since the ground-truth answers are treated as the hidden data [4] in our GEM approach.

A limitation of the stochastic modeling approach is that, on some domains, its statistical assumptions may not well-characterize crowd behavior. Accordingly, in this work we also investigate “deterministic” approaches that do not assume underlying stochastic answer generation mechanisms and hence are expected to perform robustly across a diverse set of domains. We propose two deterministic objective-based methods that jointly estimate worker intention, skill and the ground truth answers to multiclass tasks by maximizing a measure of aggregate confidence on the estimated ground truth answers measured 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 a form of overhead which limits the number of true tasks in a batch (of fixed size) that the crowd is solving. It may be expensive, time-consuming and/or impractical to devise meaningful probe tasks for a given problem domain. Accordingly, we consider our methods in both semisupervised and unsupervised settings to evaluate the gains obtained by using probes. Our experimental evaluation of the proposed schemes consisted of three levels. First, we evaluated the schemes using simulated data generated in a way consistent with our proposed stochastic generative model. This allowed us to study the robustness of our GEM algorithm by comparing estimated and actual model parameters. Second, we evaluated and compared 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 [1] and studied the comparative gains of our methods over benchmark methods on a variety of classification domains. Our final experiment involved a crowdsourcing task we posted using Amazon Mturk. Overall, we observed that our methods are especially advantageous when there is an (a priori unknown) minority of skilled workers amongst a large crowd of unskilled (as well as malicious) workers, i.e., they are able to overcome “tyranny of the masses” by estimating the workers’ intentions and skills with a reasonable level of accuracy.
II. GENERATIVE SEMISUPERVISED MODELING APPROACH

In the following framework, we separately model worker intention and skill. A worker’s intention is 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 \[\theta\] given to their answers. Here we extend malicious/adversarial worker models to multiclass tasks and hypothesize both “simple” and “crafty” adversaries.

Our approach incorporates task difficulty and worker skill explicitly and, unlike previous approaches \[21\] \[19\] \[9\], characterizes the interplay between them. 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. We also acknowledge another category of worker type known as “spammers”. These are lazy workers who simply answer randomly for all tasks. Our model well-characterizes such workers via large negative skill values.

A. 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.\(^5\) 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 \(\{T_l + 1, T_l + 2, \ldots, T_l + T_u\}\) be the index set for non-probe tasks. We assume without loss of generality that each worker is asked to solve all the tasks.\(^6\) The answers are chosen from a set \(C := \{1, 2, \ldots, K\}\). Let \(z_i \in C\) be the ground truth answer and let \(\tilde{d}_i \in (-\infty, \infty)\) represent the difficulty level of task \(i\). The (ground truth)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 \(j^{th}\) worker’s (ground truth) skill level and \(a_j\) denotes an additional degree of freedom to introduce variation in the probability mass function across workers (discussed in Section II-C). Finally the response provided to the \(i^{th}\) task by the \(j^{th}\) worker is denoted \(r_{ij} \in C\).

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.\(^7\) 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 \(\{T_l + 1, T_l + 2, \ldots, T_l + T_u\}\) be the index set for non-probe tasks. We assume without loss of generality that each worker is asked to solve all the tasks.\(^8\) The answers are chosen from a set \(C := \{1, 2, \ldots, K\}\). Let \(z_i \in C\) be the ground truth answer and let \(\tilde{d}_i \in (-\infty, \infty)\) represent the difficulty level of task \(i\). The (ground truth)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 \(j^{th}\) worker’s (ground truth) skill level and \(a_j\) denotes an additional degree of freedom to introduce variation in the probability mass function across workers (discussed in Section II-C). Finally the response provided to the \(i^{th}\) task by the \(j^{th}\) worker is denoted \(r_{ij} \in C\).

B. 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 \(C\) according to a uniform pmf \(\frac{1}{K}\).

\(^5\) The unsupervised setting is a special case where \(T_l = 0\).

\(^6\) We only make this assumption for notational simplicity. Our methodology in fact applies generally to the setting where each worker solves only a subset of the tasks.

\(^7\) The unsupervised setting is a special case where \(T_l = 0\).

\(^8\) We only make this assumption for notational simplicity. Our methodology in fact applies generally to the setting where each worker solves only a subset of the tasks.

\(^9\) One can always randomize the indexing of the answers for every task to ensure that the true answer is uniformly distributed over \(\{1, 2, \ldots, K\}\). This would remove any source of bias (e.g., toward the true answer being the first (1) or the last (K)).
2) For each worker $j \in \{1, ..., N\}$, generate $r_{ij} \in C$ for task $i$ based on the parameter-conditional probability mass function (pmf) $\beta(r_{ij}|\Lambda_{ij}, z_i)$, where $\Lambda_{ij} := \{v_j, d_j, a_j, \tilde{d}_i\}$. Also, independently for each probe task $i \in \{1, ..., T_i\}$ and each worker $j$, generate the answer $r_{ij} \in C$ based on the parameter-conditional pmf $\beta(r_{ij}|\Lambda_{ij}, z_i)$.

C. 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\footnote{The specific parametric dependence of $\beta$ on $\Lambda_{ij}$ will be introduced shortly.}. i.e., the probability that worker $j$ can solve task $i$ correctly is $\frac{1}{1+e^{-a_j(d_j - \tilde{d}_i)}}$. Note we have included a degree of freedom $a_j$ which attempts to capture the individuality of workers. It is also possible to tie this parameter, i.e., set $a_j = a$, $\forall j$.

1) Honest Workers: For an honest worker ($v_j = 1$), the pmf $\beta$ is defined as:

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

(1)

Here, the worker essentially answers correctly with high probability if $d_j > \tilde{d}_i$, and with probability $1/K$ otherwise. Note that “spammer” workers, those with $d_j << \min_i \tilde{d}_i$, will tend to answer randomly for all tasks, under this model. Next, we discuss two models for adversarial workers.

2) Simple Adversarial Workers: For the simple adversarial model, $\beta$ is given by

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

(2)

Here, essentially, the worker only chooses the correct answer (randomly) if the task difficulty defeats his skill level; otherwise he excludes the correct answer and chooses randomly from amongst the remaining answers.

3) Complex Adversarial Workers: In this case, the adversarial worker is more evasive. He answers correctly for simpler tasks with difficulty level below a certain value. Assume $\theta_j < d_j$ to be such a threshold for worker $j$. The pmf $\beta$ for this (complex) adversarial worker is given by:

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

(3)

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 ($1/K$).

In this work we will only experimentally investigate the simple model of adversarial workers.

D. Incomplete, Complete and Expected Complete Data Log Likelihood

The observed data $X = R \cup Z_C$ consists of the set $R$ of answers given by the workers to all the tasks, i.e., $r_{ij} \forall i, j$ and $Z_C = \{z_i| i \in \{1, 2, ..., T_i\}\}$ is the set of ground truth answers to the probe tasks. We express $R = R_L \cup R_U$, i.e., the union of answers to probe tasks and non-probe tasks. We choose the hidden data $[4]$ to be the ground truth\footnote{Alternative (soft) generalized step functions could also in principle be used here.}. 

$$Z = \{z_i| i \in \{1, 2, ..., T_i\}\}$$

In this case, the adversarial worker is more evasive. He answers correctly for simpler tasks with difficulty level below a certain value. Assume $\theta_j < d_j$ to be such a threshold for worker $j$. The pmf $\beta$ for this (complex) adversarial worker is given by:

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

(3)

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 ($1/K$).

In this work we will only experimentally investigate the simple model of adversarial workers.
answers to the non-probe tasks, \(i.e., Z_i, i \in \{T_i + 1, ..., T_i + T_u\}\). Based on the stochastic model in Section II-B, the incomplete data log-likelihood, which we seek to maximize in estimating \(\Lambda\), is given by

\[
\log \mathcal{L}_{inc} = \log P(\mathcal{R}, Z|\Lambda) = \log P(\mathcal{R}_L, \mathcal{R}_U, Z_L|\Lambda) = \log P(\mathcal{R}_L, Z_L|\Lambda) + \log P(\mathcal{R}_U|\Lambda)
\]

\[
= \sum_{i=1}^{T_i} \sum_{j=1}^{N} \log \frac{1}{K} \beta(r_{ij}|\lambda_{ij}, z_i) + \sum_{i=T_i+1}^{T_i+T_u} \sum_{j=1}^{N} \sum_{k=1}^{K} \log \frac{1}{K} \beta(r_{ij}|\lambda_{ij}, Z_i = k)
\]

\[
\propto \sum_{i=1}^{T_i} \sum_{j=1}^{N} \log \beta(r_{ij}|\lambda_{ij}, z_i) + \sum_{i=T_i+1}^{T_i+T_u} \sum_{j=1}^{N} \sum_{k=1}^{K} \log \beta(r_{ij}|\lambda_{ij}, Z_i = k).
\]

(4)

Treating \(Z_i, i = 1, ..., T_i\) as the hidden data within the EM framework \([4]\), the expected complete data log-likelihood, where the expectation is with respect to the pmf \(P(Z_i = k|\mathcal{X}, \Lambda)\), can be written as:

\[
\mathbb{E}[\log \mathcal{L}_c|\mathcal{X}, \Lambda] \propto \sum_{i=1}^{T_i} \sum_{j=1}^{N} \log \beta(r_{ij}|\lambda_{ij}, z_i) + \sum_{i=T_i+1}^{T_i+T_u} \sum_{j=1}^{N} \sum_{k=1}^{K} [P(Z_i = k|\mathcal{X}, \Lambda) \log \beta(r_{ij}|\lambda_{ij}, Z_i = k)]
\]

\[
= \sum_{i=1}^{T_i} \sum_{j:x_{ij}=z_i} \left[ v_j \log \left( \beta(r_{ij}|\lambda_{ij}, v_j = 1, z_i = r_{ij}) \right) + (1 - v_j) \log \left( \beta(r_{ij}|\lambda_{ij}, v_j = 0, z_i = r_{ij}) \right) \right]
\]

\[
+ \sum_{i=T_i+1}^{T_i+T_u} \sum_{j:x_{ij} \neq z_i} \left[ v_j \log \left( \beta(r_{ij}|\lambda_{ij}, v_j = 1, z_i = r_{ij}) \right) + (1 - v_j) \log \left( \beta(r_{ij}|\lambda_{ij}, v_j = 0, z_i = r_{ij}) \right) \right]
\]

\[
+ \sum_{i=T_i+1}^{T_i+T_u} \sum_{k=1}^{K} P(Z_i = k) \left[ v_j \log \left( \beta(r_{ij}|\lambda_{ij}, v_j = 1, Z_i = k) \right) + (1 - v_j) \log \left( \beta(r_{ij}|\lambda_{ij}, v_j = 0, Z_i = k) \right) \right]
\]

(5)

E. 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 \([4]\). In the expectation step, one calculates the pmf \(P(Z_i = k|\mathcal{X}, \Lambda^t)\) using the current parameter values \(\Lambda^t\), and in the maximization step, one computes \(\Lambda^{t+1} = \arg \max_{\Lambda} \mathbb{E}[\log \mathcal{L}_c|\mathcal{X}, \Lambda^t]\).

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

\[
P_t(Z_i = k|\mathcal{X}, \Lambda^t) = \frac{\prod_{j=1}^{N} \beta(r_{ij}|\lambda_{ij}^t, Z_i = k)}{\sum_{i=1}^{K} \prod_{j=1}^{N} \beta(r_{ij}|\lambda_{ij}^t, Z_i = t)}, \quad i \in \{T_i + 1, ..., T_i + T_u\}.
\]

(6)

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

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

(7)

Since \(\Lambda\) 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 (7) for our model. Instead, we use a generalized M-step approach \([11] \) \([7]\) to iteratively maximize over the two parameter subsets \(\{v_j \neq j\}\), and \(\{\{d_{ij}, a_{ij}\} \neq j\}, \{\bar{d}_i \neq i\}\).

M1 Substep: Since (7) is an additive function of terms that each depend on a single variable \(v_j\), we can find a closed form solution for \(v_j \neq j\) given all other parameters fixed:

\[
\tilde{v}_j = \arg \max_{v_j \in (0,1)} \mathbb{E}(\log \mathcal{L}_c(\{v_j\})|\mathcal{X}, \tilde{\Lambda}\{\bar{v}_j\})
\]

(8)
Here \( X_j \) is the set of answers provided by the \( j \)th worker and the ground truth answers for the probe tasks that he answered and \( \hat{\Lambda} \) is the result of the previous M2 substep.

**M2 Substep:** We maximize \( \mathbb{E}[\log L_c(\Lambda \setminus \{v_j\} | X, \{\tilde{v}_j\})] \) with respect to \( \Lambda \setminus \{v_j\} \) given \( \{\tilde{v}_j\} \) fixed from the previous M1 substep. For this, we use a gradient ascent algorithm which ensures monotonic increase in \( \log L_{inc} \), but which may only find a local maximum, rather than a global maximum of \( \mathbb{E}[\log L_c(\Lambda \setminus \{v_j\} | X, \{\tilde{v}_j\})] \). At convergence, the result is stored in \( \hat{\Lambda} \setminus \{v_j\} \). The M1 and M2 substeps are applied alternately, iteratively, until convergence. \( \Lambda^{t+1} \) stores the result of the generalized M-step at convergence.

**Inference:** Note that the E-step (6) computes the \( a \ posteriori \) probabilities of ground-truth answers. Thus, after our GEM learning has converged, a maximum \( a \ posteriori \) decision rule applied to (6) gives our crowd-aggregated estimates of the true answers for the non-probe tasks.

### F. Unsupervised GEM

Note that when probe tasks are not included in the batch of tasks, an unsupervised specialization of the above GEM algorithm is obtained. In particular, we have \( T_1 = 0 \), with the first term in (4) and the first two terms in (5) not present. Our above GEM algorithm is accordingly specialized for this case. In Section IV we will evaluate the unsupervised GEM based scheme along with all other methods.

#### III. Energy-Constrained Weighted Plurality Aggregation

Performance of our GEM approach will in general depend on how well the true answer generation mechanism resembles the one assumed in Section II-B. We would also therefore like to explore an alternative “principle” on which to base crowd aggregation, without any explicit assumption about the underlying stochastic answer generation 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 semisupervised settings, but for clarity of presentation, we will focus on the unsupervised setting and then delineate how to extend these approaches to exploit probe tasks, if available.

#### A. From simple plurality to weighted plurality voting

We will introduce and define new variables to clearly explain our approach. Let \( T = T_i + T_u \) and \( \hat{z}_j \) be the \( K \times 1 \) vector representing the inferred ground truth answers with \( \hat{z}_{im} \in \{0, 1\} \) and \( \sum_m \hat{z}_{im} = 1 \), i.e., \( \hat{z}_{im} \) is 1 when the inferred answer to the \( i \)th task is \( m \). Also, \( \hat{Z} = (\hat{z}_i, i = 1, ..., T) \). All other definitions from section II-A will be used. A natural extension of majority voting to the multicategory case is plurality voting, where the answer that gets the maximum number of votes is the inferred answer for that task. 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\(^{12}\)

\[
\max_{\hat{Z}} \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} \sum_{j=1}^{N} \delta(r_{ij} - m) \tag{9}
\]

subject to

\[
\hat{z}_{im} \in \{0, 1\}, \quad \sum_{m}^{K} \hat{z}_{im} = 1, \quad i = 1, ..., T,
\]

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

Plurality-based voting is expected to perform well when all the workers are honest and “equally reliable”, with worker accuracy greater than that of a spammer.\(^{13}\) However, the crowd may consist of heterogeneous workers with varying skill levels and intentions, as we considered before in Section II. For the most challenging but realistic “tyranny of the masses” case where a small proportion of highly skilled workers exist among a mass of unskilled

---

\(^{12}\)Ties could be broken by randomly selecting from among the set of plurality answers.

\(^{13}\)The expected accuracy of the spammer is \( \frac{1}{K} \), where \( K \) is the number of possible answers for each task.
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 plurality 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. In order to ensure well-posed optimization problems, we will impose an energy constraint on the weight vector. We will first propose to jointly estimate the worker weights and the ground truth answers consistent with solving the following optimization problem:

$$\max_{Z, w} \psi_{wp}(w, Z) = \sum_{j=1}^{N} w_j \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} \delta(r_{ij} - m)$$  \hspace{1cm} (10)$$

subject to

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

Here, we maximize the average weighted plurality score (a measure of aggregate confidence in decisionmaking). (10) is a non-convex optimization problem, whose (locally optimal) solutions we will obtain 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 convergence.

**Step 1**: For fixed $w$ and for each task $i$, the choice for $\hat{z}_i$ which maximizes (10) is:

$$\hat{z}_{im} = \begin{cases} 1 & \text{if } \sum_{j: r_{ij} = m} w_j > \sum_{j: r_{ij} = k} w_j \forall k \neq m \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (11)$$

**Step 2**: Given fixed $\hat{Z}$ we compute the optimum $w$, maximizing (10). The Lagrangian for this optimization problem is

$$\mathcal{L} = \sum_{j=1}^{N} w_j \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} (\delta(r_{ik} - m) + \lambda \sum_{j=1}^{N} w_j^2 - 1).$$ \hspace{1cm} (12)$$

Differentiating with respect to $w_k$, we get

$$\frac{\partial \mathcal{L}}{\partial w_k} = \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} \delta(r_{ik} - m) + 2\lambda w_k = 0$$ \hspace{1cm} (13)$$

$$\Rightarrow w_k = -\frac{1}{2\lambda} \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} \delta(r_{ik} - m) \forall k$$

We can compute $\lambda$ by squaring the above and then summing over all workers $j$, to find

$$\lambda = -\frac{1}{2} \sqrt{\frac{\sum_{j=1}^{N} \left( \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} (\delta(r_{ij} - m) \right)^2}{\sum_{j=1}^{N} \left( \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} (\delta(r_{ij} - m) \right)^2}}$$ \hspace{1cm} (14)$$

Hence the optimal value of $w_k$ is given by

$$w_k^* = \frac{\sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} (\delta(r_{ik} - m)}{\sqrt{\sum_{j=1}^{N} \left( \sum_{i=1}^{T} \sum_{m=1}^{K} \hat{z}_{im} (\delta(r_{ij} - m) \right)^2}} \forall k.$$ \hspace{1cm} (15)$$
Each of the above two steps ascends in the objective (reward) function $\psi_{wp}$, with convergence to a local maximum, starting from an initial weight vector $w = \epsilon \mathbf{1}$, where $\epsilon = \frac{1}{\sqrt{N}}$ and $\mathbf{1}$ is an $N \times 1$ vector of all ones.

B. Accounting for Adversaries

Note that we have not accounted for adversarial workers in (10). To do so now, suppose that an adversarial worker $k$ 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.

1) Introducing Binary Parameters to Represent Intention: Suppose we introduce an additional set of variables $\mathbf{v}$, 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:

$$
\max_{Z,w,v} \psi_{bp}(w, \hat{Z}, v) = \sum_{j=1}^{N} \sum_{i=1}^{T} \sum_{m=1}^{K} (v_j \hat{z}_{im} w_j (r_{ij} - m) + \frac{1}{K-1} (1 - v_j) \hat{z}_{im} w_j (1 - \delta(r_{ij} - m)))
$$

subject to

$$
\sum_{j=1}^{N} w_j^2 = 1, \quad \hat{z}_{im} \in \{0,1\}, \quad \sum_{m=1}^{K} \hat{z}_{im} = 1, \quad v_j \in \{0,1\}.
$$

Here, when a worker is identified as adversarial, we allocate equal weight ($\frac{w_j}{K-1}$) to all the answers except the one the worker has chosen.\(^{14}\) A locally optimal algorithm, maximizing the objective $\psi_{bp}$ starting from an initial weight vector $w = \epsilon \mathbf{1}$, consists of the following three iterated steps:

**Step 1:** For fixed values of $w$ and $v$ and for each task $i$, the optimal $\hat{z}_i$, maximizing (16), is chosen as:

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

**Step 2:** For fixed values of $\hat{Z}$ and $v$, the optimal $w$, maximizing (16), is given by:

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

**Step 3:** For fixed values of $\hat{Z}$ and $w$, the optimal $v$, maximizing (16), is:

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

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 (18) will always be non-negative, 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” associated with the previous algorithm. First, let us note that negative weights can be used to signify adversarial workers, and treated accordingly. Thus

\(^{14}\)The weights across incorrect answers are normalized by $\frac{1}{K-1}$ so that each worker’s contribution to a given task equals his weight $w_j$.\)
rather than apportion \( \frac{w_j}{K-1} \) when \( v_j = 0 \), as in Section [III-B1] we can equivalently apportion \( \frac{|w_j|}{K-1} \) when \( w_j < 0 \) (and, as we shall see shortly, avoid the need to explicitly introduce binary intention parameters). Second, to appreciate a possible alternative to \( i \), suppose an influential nonadversarial worker \( (w_j \text{ large and positive}) \) does not agree with the (current) weighted plurality decision for a given task. Rather than \( \text{not} \) contributing to the plurality score and the objective function, this worker could \( \text{subtract} \) his weight from the weighted plurality score (in fact, from the scores of all answers with which he disagrees) to try to \( \text{alter} \) the weighted plurality decision. Interestingly, we can achieve \( \text{both} \) of these mechanisms, and also avoid including binary intention parameters, with the introduction of a \( \text{single} \) additional cost term, thus modifying the objective function from Section [III-B1]. Specifically, supposing that \( \hat{Z} \) is the current estimate of ground truth answers, then worker \( j \)'s contribution to the objective function is now taken to be:

\[
w_j \sum_{i=1}^{T} \sum_{m=1}^{K} (\hat{z}_{im} \delta(r_{ij} - m) - \frac{1}{K-1} \hat{z}_{im}(1 - \delta(r_{ij} - m)))
\]  

(20)

Let us consider the two cases \( w_j < 0 \) and \( w_j > 0 \). If \( w_j < 0 \), then in (20) the adversary's weight magnitude is (fully) subtracted from the current plurality score if his answer agrees with that of the current plurality. On the other hand, if his answer disagrees with the current plurality, his weight magnitude is equally apportioned amongst the remaining answers. Thus, (20) accounts for adversarial workers precisely as we intend. Next, suppose that \( w_j > 0 \). We can see that (20) behaves as desired \( \text{in this case as well} \), fully adding \( w_j \) to the plurality score if he agrees with the plurality decision, and, if he disagrees with the plurality, subtracting \( \frac{w_j}{K-1} \) from the scores of the answers with which he disagrees. Thus (20) accounts for \( \text{both} \) adversarial and non-adversarial workers precisely in the way we intend. Accordingly, we modify the objective function using (20) as the per-worker contribution:

\[
\max_{\hat{Z},\mathbf{w}} \psi_{\text{neg}}(\mathbf{w}, \hat{Z}) = \sum_{j=1}^{N} w_j \left[ \sum_{i=1}^{T} \sum_{m=1}^{K} (\hat{z}_{im} \delta(r_{ij} - m) - \frac{1}{K-1} \hat{z}_{im}(1 - \delta(r_{ij} - m))) \right]
\]

subject to

\[
\sum_{j=1}^{N} w_j^2 = 1, \quad \hat{z}_{im} \in \{0, 1\}, \quad \sum_{m=1}^{K} \hat{z}_{im} = 1.
\]

We further note that, supposing \( \hat{Z} \) are the ground truth answers, then the per worker term bracketed in (21) \( \sum_{i=1}^{T} \sum_{m=1}^{K} (\hat{z}_{im} \delta(r_{ij} - m) - \frac{1}{K-1} \hat{z}_{im}(1 - \delta(r_{ij} - m))) \) 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. Consequently, assigning a non-zero weight to a spammer is clearly sub-optimal due to the energy constraint on the weight vector, \( \mathbf{i.e.} \), to maximize (21), spammers will (asymptotic in \( T \)) be assigned zero weights. Thus (21) properly accounts for honest workers, malicious workers and spammers.

Our locally optimal algorithm for (21) consists of iteration of the following two steps, starting from the weight vector initialization:

**Step 1:** For a fixed \( \mathbf{w} \) and for each task \( i \), choose \( \hat{z}_{im} \) as

\[
\hat{z}_{im} = \begin{cases} 
1 & \text{if } \sum_{j:x_{ij}=m} w_j - \frac{1}{K-1} \sum_{j:x_{ij} \neq m} w_j > \sum_{j:x_{ij}=k} w_j - \frac{1}{K-1} \sum_{j:x_{ij} \neq k} w_j \quad \forall \ k \neq m \\
0 & \text{otherwise}
\end{cases}
\]

(22)

**Step 2:** For fixed \( \hat{Z} \), the optimal \( \mathbf{w} \) is:

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

(23)
C. Semisupervised Case

Note that for both the methods in Sections III-B1 and III-B2, 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.

IV. Experiments

Experiments were performed using synthetic data as well as data generated by a crowdsourced multiclass 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 of the simple type in all our experiments. In addition to the proposed schemes, we also compared with simple (multiclass) plurality voting and its semisupervised version, which exploits the probe tasks. Let us reiterate the different methods that we will apply and study in this section:

i) PLU: Simple plurality.
ii) SS-PLU: Semisupervised plurality.
iii) US-SW: Unsupervised objective-based method that uses binary parameter for intention described in Section III-B1.
iv) SS-SW: Semisupervised specialization of US-SW (using probe tasks).
v) US-NEG: Unsupervised objective-based method without intention parameters, described in Section III-B2.
vi) SS-NEG: Semisupervised specialization of US-NEG (using probe tasks).
vii) SS-GEM: Semisupervised GEM based scheme described in Section II.
viii) US-GEM: Unsupervised specialization of SS-GEM (without any probe tasks).

A. 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 II-B and II-C. 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 \( d_j \sim \mathcal{N}(1, 400), \ a_j \sim \mathcal{N}(0.3, 0.2) \); 10% of workers were adversarial. The tasks were generated with \( \tilde{d}_i \sim \mathcal{N}(8, \sigma^2) \), where \( \sigma^2 \) 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 variance in 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 I. Table II 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. We also see in Figure 1 the high correlation between the estimated and actual values of worker skills and task difficulties. Also noteworthy from Table I is the superior performance of US-GEM over the other unsupervised objective-based schemes. We plot in Figure 2 the histogram of worker accuracies for one of the trials for the first entry in Table I. This illustrates that highly skilled workers represent a small minority in these experiments.

In the second part we separately evaluated the deterministic objective-based schemes (US-SW and US-NEG) and compared them with simple plurality voting. 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 III shows comparison across varying proportions of spammers and adversarial workers in the crowd. For the second model, we generated the data according to the stochastic model in Section I-B with the task difficulty for each task \( \tilde{d}_i \sim \mathcal{U}[0, 8] \). We created three category 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 IV shows the comparison of the schemes for this model of synthetic data. Note that for both the experiments in Tables III and IV we averaged the number of erroneous tasks over 20 trials. We can observe from Tables III and IV that US-NEG clearly outperforms US-SW and PLU.

\[^{15}\text{This is weighted plurality, where each worker is weighted by the fraction of probe tasks that he answered correctly.}\]
| Task Variance | Average erroneous tasks |
|---------------|-------------------------|
|               | PLU | SS-PLU | US-SW | US-NEG | SS-GEM | US-GEM |
| 4000          | 23.4| 22.1   | 21.71 | 22.42  | 16.9   | 19.85  |
| 2000          | 21.2| 20     | 20.2  | 20.5   | 14.6   | 15.2   |
| 1000          | 19.2| 16.2   | 18.2  | 16.5   | 9.6    | 9.8    |
| 500           | 11.1| 8.1    | 8.42  | 7.85   | 3.14   | 4.14   |
| 250           | 5.85| 5.8    | 5.28  | 5.28   | 2.14   | 2.57   |

TABLE I: Synthetic data generated using the stochastic model: Changing task difficulty variance

| Assignment degree | Average erroneous tasks |
|-------------------|-------------------------|
|                   | PLU | SS-PLU | US-SW | US-NEG | SS-GEM | US-GEM |
| 20                | 31.28| 31.6   | 31.2  | 29.85  | 27     | 28     |
| 40                | 22.85| 21.2   | 21.42 | 21.14  | 18.42  | 18.85  |
| 60                | 19.42| 18.6   | 18    | 18     | 14.28  | 15.42  |
| 80                | 16.57| 16     | 15.4  | 15.14  | 14.14  | 13.4   |

TABLE II: Synthetic data generated using the stochastic model: Changing number of worker assignments.

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 additional layer of optimization 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 $v$ initialized based on the US-NEG solution (with the sign of $w_j$ used to initialize $v_j$). We can see from Table III that US-HYB certainly performs better than US-SW. Table V shows the number of trials (out of 100) when both i) US-HYB gave fewer decision errors than US-SW and ii) the objective function value using US-HYB was strictly greater than using US-SW. We can see from the entries in this table and Table III that there is a strong correlation between US-HYB achieving a greater objective function value and US-HYB achieving fewer decision errors than US-SW.

B. 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 [14]. Each such classifier (worker) is a strong learner obtained by applying multiclass boosting.
Fig. 2: Histogram of worker accuracies for high variance regime.

| Percentage adversarial workers | Number of erroneous tasks | US-SW/US-NEG/US-HYB/PLU |
|-------------------------------|---------------------------|--------------------------|
|                               |                           | Percentage spammers      |
|                               |                           | 10  | 20  | 30  | 40  | 50  | 60  |
| 0                             |                           | 5.4/5.1/6.7/8.7/         | 6.0/5.7/9.1/6.6/       | 10.5/9.05/9.1/9.6/    | 10.9/9.9/9.7/16.2/  | 13.7/11.7/11.4/22.3 |
| 5                             |                           | 5.6/5.4/6.6/6.4/9.8/    | 8.7/8.6/10.1/9.3/      | 12.8/11.9/12.0/       | 19.2/17.1/16.9/29.5 |
| 10                            |                           | 6.8/6.1/6.6/6.6/11.9/  | 8.8/8.0/10.3/9.4/      | 13.8/12.4/12.3/       | 29.7/16.2/16.3/30.7 |
| 15                            |                           | 5.8/5.0/8.4/7.3/7.7/14.4 | 10.9/9.8/13.5/12.5/    | 16.6/14.8/24.3/       | 21.0/21.3/21.6/37.8 |
| 20                            |                           | 7.5/13.8/8.3/7.5/13.8/  | 12.2/11.2/13.7/11.8/   | 18.8/16.8/29.2/       | 22.3/22.9/22.0/44.8 |

TABLE III: Synthetic data: Comparing objective-based methods for spammer, hammer and adversary model

| Percentage high skilled adversarial workers | Number of erroneous tasks | PLU/US-SW/US-NEG |
|--------------------------------------------|---------------------------|-----------------|
|                                            |                           | Percentage low skilled honest workers |
|                                            |                           | 10  | 20  | 30  | 40  | 50  | 60  |
| 5                                          |                           | 9.4/6.7/6.0/6.0/         | 12.7/6.9/6.7/9.6/      | 14.4/8.3/7.2/         | 20.3/11.3/10.5/     | 26.9/17.3/15.4     |
| 10                                         |                           | 9.8/6.9/6.1/6.1/         | 15.8/8.6/7.7/          | 18.4/10.5/9.8/        | 23.1/13.2/12.6/     | 29.5/18.3/16       |
| 15                                         |                           | 10.5/6.0/5.6/7.1/        | 16.7/9.4/7.7/11.3/     | 22.9/12.8/13.7/       | 37/22.4/20.1       |
| 20                                         |                           | 12.8/6.7/6.4/8.5/        | 20.4/9.1/8.4/12.6/     | 25.7/14.7/18.2/       | 42.5/27.4/22        |

TABLE IV: Synthetic data: Comparing objective-based methods for data generated using model in Section II-B
TABLE V: Comparing US-SW and US-HYB

| Percentage Adversarial | Number of trials |
|------------------------|-----------------|
|                        | Percentage Spammers |
|                        | 10   | 20   | 30   | 40   | 50   | 60   |
| 0                      | 34   | 42   | 34   | 67   | 63   | 45   |
| 5                      | 17   | 47   | 49   | 54   | 62   | 76   |
| 10                     | 24   | 57   | 48   | 49   | 64   | 77   |
| 15                     | 71   | 31   | 62   | 57   | 81   | 95   |
| 20                     | 55   | 41   | 45   | 66   | 91   | 94   |

We ran Multiboost \cite{2} 100 times to create a crowd of 100 workers for four domains that are realistic as crowdsourcing domains: Pen Digits, Vowel, Dermatology, and Nominal. 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 of the data samples from the domain were used for training the strong (ensemble) classifiers/workers. The average of the number of crowd-aggregated erroneous tasks was computed across 5 trials, where each trial consisted of a freshly generated crowd of workers and set of tasks. In Tables VI, VII, VIII, and IX we give performance evaluation 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% 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 3.2. In Table VI the worker accuracy mean and variance across all workers is based on the number of correctly answered tasks (both probe and non-probe) for each worker computed in the absence of adversaries.

We can see in Table VI for the Pen Digits dataset, the gain in performance with 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) perform better than other schemes for relatively higher means of worker accuracy. Figures 3 and 4 show the histogram of worker accuracies for low and high mean cases, respectively. In Figure 3 we plot the histogram of worker accuracies corresponding to the first entry in Table VI (for a single trial). We can observe an extremely skewed distribution of worker accuracies where a tiny minority of extremely high skilled workers exist amidst others who are mostly spammers. This distribution seems to best “agree” with the GEM based schemes. Figure 4 corresponds to the last entry in Table VI. Here we observe a less skewed distribution of worker accuracies. The objective based US-NEG and SS-NEG schemes perform better in this regime. The GEM based scheme is able to exploit simple adversaries much more than other methods, achieving improved crowd aggregation accuracy compared to the case where no adversaries are present, for almost all the entries in Table VI. Also observed from the table on this dataset, 10% 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 when probe tasks are available. 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”. As seen in Table VII on Dermatology, although unsupervised, US-NEG performs very close to SS-GEM when adversarial workers are absent. However, its performance degrades when adversaries are introduced in the crowd. SS-GEM greatly outperforms other methods. Also note that, on

\footnote{We resampled the dataset to have only odd digits.}

\footnote{Hungarian named entity dataset \cite{15}. Identifying and classifying proper nouns into four categories: not a proper noun, person, place, and organization.}
### TABLE VI: Experiments using Pen Digits dataset

| 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 |
| 24.19                | 262.8                   | 24.23                  | 23.8 | 11     | 11.2  | 11.1  | 10.8   | 10.8   | 9.4    | 8.4    |
| 26.27                | 294.2                   | 32.5                   | 22   | 10.4   | 11    | 11    | 12     | 11.8   | 12     | 8.6    | 8.6    |
| 29.5                 | 465.5                   | 42.9                   | 17.6 | 11.8   | 12    | 11.9  | 10.8   | 10.8   | 9.5    | 9.2    |
| 33.9                 | 485.6                   | 88.8                   | 14.8 | 8.8    | 9.4   | 9.4   | 7.2    | 7      | 9.6    | 9      |
| 38.8                 | 522.6                   | 182.2                  | 14   | 7.6    | 7.8   | 7.8   | 6      | 6      | 8.2    | 7.8    |
|                      |                          |                        | 16.4 | 7.8    | 8.2   | 8.2   | 6.6    | 6.6    | 8.2    | 8      |

### TABLE VII: Experiments using Dermatology dataset

| 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 |
| 19.7                 | 215.1                    | 44.1                   | 48  | 23.2   | 23   | 22.9  | 15.8   | 15.8   | 24.2   | 8.6    |
| 22.6                 | 399.8                    | 66.3                   | 40.2| 12.2   | 13   | 12.8  | 6      | 6      | 20.4   | 8.2    |
| 26.9                 | 480.3                    | 76.6                   | 50.4| 11.2   | 24.6 | 24.2  | 19.2   | 19.2   | 17.4   | 7      |
| 27.6                 | 563.1                    | 88.9                   | 26.8| 8.6    | 8.4  | 8.3   | 7      | 6.8    | 7      | 6.8    |
| 31.7                 | 632.1                    | 110.3                  | 14.4| 4      | 4.4  | 4.4   | 4.8    | 4.8    | 4      | 4      |
|                      |                          |                        | 19  | 5      | 4    | 4     | 4.3    | 4.2    | 3.8    | 3.6    |

This dataset, unlike Pen Digits, probe supervision greatly assists the inference using GEM for lower mean worker accuracy (as seen in the first two entries of Table VII). In Figures 5 and 6, we plot the worker accuracy profiles for two cases: where probe supervision is and is not greatly beneficial, respectively. Table VIII shows the results on the vowel dataset. Here, US-NEG and SS-NEG clearly outperform all other methods, even SS-GEM, in the absence of adversaries. US-GEM and SS-GEM, again, perform better than all others when adversarial workers are introduced. Figure 7 shows the distribution of worker accuracies in the crowd in a histogram for one of the trials corresponding to the lowest mean accuracy in Table VIII. We observed from our experiments with the Nominal dataset (Table IX) that the GEM based methods perform better than others overall, although the performance gap diminishes as the mean worker accuracy improves.

### C. MTurk Experiment

We designed an image labeling task where workers had to provide the country of origin, choosing from *Afghanistan, India, Iran, Pakistan, Tajikistan*. Some of the regions in these countries look 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 between the two corresponding nodes. Table X shows the average number of erroneous crowd-aggregated answers for the methods under study as we varied the...
### TABLE VIII: Experiments using Vowel dataset

| Worker accuracy mean | Worker accuracy variance | Task accuracy variance | Without adversarial workers | With 10% adversarial workers |
|----------------------|--------------------------|------------------------|----------------------------|----------------------------|
|                      |                          |                        | PLU | US-SL | SS-SW | US-SW | US-NEG | SS-NEG | US-GEM | SS-GEM |
| 24.2                 | 390.1                    | 35.6                   | 25.6 | 15.8  | 15.8  | 15.8  | 13.4   | 13.5   | 16.2   | 15.6   |
| 26.8                 | 412.4                    | 42.3                   | 23.1 | 14.2  | 14.7  | 14.7  | 13.4   | 13.2   | 15.9   | 15.7   |
| 29                   | 445.8                    | 57.4                   | 22.5 | 13.8  | 13.9  | 13.9  | 12.8   | 12.8   | 15.6   | 15.2   |
| 32.2                 | 624.6                    | 81.8                   | 17.8 | 12.2  | 12.4  | 12.4  | 11     | 10.8   | 13.6   | 13.6   |
| 35.9                 | 716.4                    | 118.9                  | 16.4 | 13.6  | 13.4  | 13.3  | 13.2   | 13.2   | 16.4   | 15.6   |

TABLE IX: Experiments using Nominal dataset

| Worker accuracy mean | Worker accuracy variance | Task accuracy variance | Without adversarial workers | With 10% adversarial workers |
|----------------------|--------------------------|------------------------|----------------------------|----------------------------|
|                      |                          |                        | PLU | US-SL | SS-SW | US-SW | US-NEG | SS-NEG | US-GEM | SS-GEM |
| 45.4                 | 575.4                    | 339.1                  | 19  | 18.8  | 18.8  | 18.8  | 18.2   | 18.2   | 18.4   | 18.2   |
| 49.1,3               | 599.2                    | 328.6                  | 10.4| 8.4   | 9.2   | 9.2   | 9.1    | 8.1    | 8.1    | 7.7    |
| 51.3                 | 649.9                    | 325.1                  | 9.8 | 7.8   | 8     | 8     | 7.8    | 7.6    | 7.7    | 5.6    |
| 52.6                 | 678.5                    | 298.5                  | 9.6 | 6.2   | 6.7   | 6.7   | 5.3    | 5.3    | 5.1    | 4.7    |
| 54.3                 | 878.1                    | 233.1                  | 4.2 | 3.2   | 3.2   | 3.1   | 3.2    | 3.2    | 2.9    | 2.6    |

Fig. 3: Pen Digits dataset: Histogram of worker accuracies with a skewed distribution of skills
Fig. 4: Pen Digits dataset: Histogram of worker accuracies with a more gradual “spread” of skill distribution

Fig. 5: Dermatology dataset: Histogram of worker accuracies. Supervision greatly helps

Fig. 6: Dermatology dataset: Histogram of worker accuracies. Supervision does not help much.
number of tasks assigned to each worker. The average was computed over 5 trials, each time generating a new instance of a random bipartite graph and using 5 randomly chosen probe tasks. The histogram of worker accuracies is shown in Figure 9. From the histogram we can observe that only a small fraction of workers could answer most tasks correctly.

V. RELATED WORK

Crowdsourcing has attracted much interest from the research community and it has been studied in different forms and under various scenarios and assumptions. In this section, however, we will focus only on those contributions that are closely related to our work. Stochastic models for the generation of workers’ answers have been previously considered in [3], [13], [21], [19]. In [3] the parameters of the model are the per-worker confusion matrices that are jointly estimated with the ground truth answers. This method was extended in [22], where a distinct probability distribution was given over the answers for every task-worker pair. But [22] does not consider a per-task parameter representing task difficulty. In our model, the distribution over answers for a task explicitly depends on the task
difficulty that is essentially “perceived” by workers attempting to solve it. More specifically, we explicitly model the interplay between worker skill and task difficulty. Our focus in this paper has been mostly on crowdsourcing tasks that require an expertise (skill) present only in the minority of the crowd. In [21] and [19], task difficulty was considered explicitly, but only for the binary case. [10] considered task difficulty for ordinal data, wherein a higher difficulty level adds more variance to the distribution of workers’ elicited answers without affecting the mean (which equals the ground truth value). Our method of incorporating task difficulties is novel, as we use them in a comparative paradigm with worker skill in our soft threshold-based model. Unlike [19], which assumes all tasks are drawn from the same classification domain, our approach is applicable even when the batch of tasks is heterogeneous, i.e., not necessarily all drawn from the same (classification) domain. This is due to the fact that our model does not consider domain-dependent features [19] and also because our model is invariant to task-dependent permutations applied to the answer space. Note that [12] allows biasing towards certain answers, which only applies when all the tasks belong to the same classification domain.

Adversarial workers in the binary case were accounted for in [21] and [9]. In this work, we characterized adversarial behavior for a more generalized (multicategory) setting and proposed several realistic adversarial models. We also showed how we can retain the interpretation of negative weights as representing adversaries in generality from the binary [9] to the multicategory case. Moreover, we showed that our approach exploits responses from (simple) adversaries to actually improve the overall performance. [9] and [8] consider other statistical methods, such as correlation-based rules and low rank approximation of matrices. 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, [17] have tried to extend the low rank approximation approach to the case when the tasks do not have a ground truth answer and the answers (from a categorical space) can be subjective. In this case, “schools of thought” are discovered via clustering and the average size of clusters for each task is representative of its ease (clarity).

VI. CONCLUSION AND FUTURE WORK

This paper develops two novel paradigms to approach crowd aggregation and inference in multicategory crowdsourcing tasks: a generative stochastic modeling-based approach that explicitly models worker skill, intention and task difficulty and an objective-based non-generative approach that extends the notion of plurality voting to weighted plurality and also incorporates worker’s intention. These approaches are specifically designed to overcome “tyranny of the masses” by identifying the highly skilled workers in the crowd. We dissect a worker’s accuracy into skill and intention, which allows us to extend the notion of adversarial behavior to a multiclass scenario. Along with experiments on synthetic and real world data, we evaluated our schemes using a heterogeneous crowd of simulated workers, where each worker was an automated classifier generated via boosting. From our experiments, we verified that our proposed schemes outperform other benchmark schemes, especially in the case when a small minority of highly skilled workers is present in a crowd of mostly low skilled workers. Moreover, our schemes were able to detect and leverage adversarial behavior. In the future, we would like to comprehensively evaluate more complex adversarial models, including the one proposed in this paper. We would also like to explore the possibility of different attacks on crowdsourcing systems, for instance 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, depending on the size of the group. We will also study the relationship between our objective-based schemes and low rank approximation schemes and attempt to extend message passing schemes to a multicategory setting.
REFERENCES

[1] K. Bache and M. Lichman. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences, 2013.

[2] D. Benbouzid, R. Busa-Fekete, N. Casagrande, F.D. Collin, B. Kégl, et al. Multiboost: a multi-purpose boosting package. Journal of Machine Learning Research, 13:549–553, 2012.

[3] A.P. Dawid and A.M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. Applied Statistics, pages 20–28, 1979.

[4] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B (Methodological), pages 1–38, 1977.

[5] J. R. Douceur. The sybil attack. In Peer-to-peer Systems, pages 251–260. 2002.

[6] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-riding and whitewashing in peer-to-peer systems. In ACM SIGCOMM Workshop on Practice and Theory of Incentives in Networked Systems, pages 228–236, 2004.

[7] M. W. Graham and D. J. Miller. Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection. IEEE Transactions on Signal Processing, 54(4):1289–1303, 2006.

[8] D. Karger, S. Oh, and D. Shah. Budget-optimal crowdsourcing using low-rank matrix approximations. In 49th IEEE Annual Allerton Conference on Communication, Control, and Computing, pages 284–291, 2011.

[9] D.R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. Advances in Neural Information Processing Systems, 2011.

[10] B. Lakshminarayanan and Y. W. Teh. Inferring ground truth from multi-annotator ordinal data: a probabilistic approach. arXiv.org preprint arXiv:1305.0015, 2013.

[11] X. L. Meng and D. Van Dyk. The EM algorithm old folk-song sung to a fast new tune. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 59(3):511–567, 1997.

[12] V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. Journal of Machine Learning Research, 13:491–518, 2012.

[13] V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. Journal of Machine Learning Research, 11:1297–1322, 2010.

[14] R. E. Schapire. The strength of weak learnability. Machine learning, 5(2):197–227, 1990.

[15] U. Syed and B. Taskar. Semi-supervised learning with adversarially missing label information. Advances in Neural Information Processing Systems, 2010.

[16] G. Szarvas, R. Farkas, and A. Kocsor. A multilingual named entity recognition system using boosting and c4. 5 decision tree learning algorithms. Discovery Science, pages 267–278, 2006.

[17] Y. Tian and J. Zhu. Learning from crowds in the presence of schools of thought. In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 226–234, 2012.

[18] M. Vukovic. Crowdsourcing for enterprises. In IEEE World Conference on Services-I, pages 686–692, 2009.

[19] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. Advances in Neural Information Processing Systems, 6(7):8, 2010.

[20] P. Welinder and P. Perona. Online crowdsourcing: rating annotators and obtaining cost-effective labels. In Proc. IEEE Conf on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 25–32, 2010.

[21] 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. Advances in Neural Information Processing Systems, 22:2035–2043, 2009.

[22] D. Zhou, J. Platt, S. Basu, and Y. Mao. Learning from the wisdom of crowds by minimax entropy. Advances in Neural Information Processing Systems, pages 2204–2212, 2012.