Debiassing Crowdsourced Quantitative Characteristics in Local Businesses and Services

Robin Wentao Ouyang, Lance Kaplan, Paul Martin, Alice Toniolo, Mani Srivastava and Timothy J. Norman

1 University of California, Los Angeles, CA, USA
2 Army Research Lab, USA
3 University of Aberdeen, Aberdeen, UK
wouyang@ucla.edu, lkaplan@ieee.org, pdmartin@ucla.edu, a.toniolo@abdn.ac.uk, mbs@ucla.edu, t.j.norman@abdn.ac.uk

ABSTRACT

Information about quantitative characteristics in local businesses and services, such as the number of people waiting in line in a cafe and the number of available fitness machines in a gym, is important for informed decision, crowd management and event detection. In this paper, we investigate the potential of leveraging crowds as sensors to report such quantitative characteristics and investigate how to recover the true quantity values from noisy crowdsourced information. Through experiments, we find that crowd sensors have both bias and variance in quantity sensing, and task difficulties impact the sensing accuracy. Based on these findings, we propose an unsupervised probabilistic model to jointly assess task difficulties, ability of crowd sensors and true quantity values. Our model differs from existing categorical truth finding models as ours is specifically designed to tackle quantitative truth. In addition to devising an efficient model inference algorithm in a batch mode, we also design an even faster online version for handling streaming data. Experimental results in various scenarios demonstrate the effectiveness of our model.

Categories and Subject Descriptors
H.4 [Information Systems Applications]: Miscellaneous

Keywords
Crowdsourcing, humans as sensors, truth discovery, probabilistic graphical models

1. INTRODUCTION

Nowadays, users mainly rely on web search or past experience for making decisions on which local business or service to visit and when to. Such information usually consists of business ratings, reviews, distance, and empirical service speed. No real-time information is available about the current quantitative characteristics such as the number of people waiting in a bank, the number of available tables in a restaurant and the number of available fitness machines in a gym. Such information has the potential to enable well-informed decision for end users and timely crowd management for business managers and service providers.

Consider the scenario of determining when to visit the customer service at a local bank as an example. Without knowing how many people are currently waiting to be serviced, Alice is likely to visit the bank at any time when she is available, encountering a large crowd and a long waiting time. On the other hand, when such information is available, Alice can make better informed decision about whether to visit the customer service right now or at a time when the line is expected to be shorter. Moreover, such information can enable subscription services (e.g., sending a notification to a user when an expected condition happens), new search experience (e.g., searching local businesses or services according to their ambience characteristics [25]), and timely crowd management (e.g., opening more service windows or adding more staff when the waiting line is expected to be long [26]). Furthermore, as such information accumulates over time, it can enable pattern discovery and forecasting service based on the analysis of historical data.

In this paper, we investigate the potential of leveraging crowds as sensors to sense such quantitative characteristics (Figure 1) and investigate how to recover the true quantity values from noisy crowdsourced information. When a crowd participant visits a local business or a service, she has the opportunity to observe and estimate the characteristics of interest (e.g., number of people waiting in line). As crowds contribute observations throughout a day, such quantitative characteristics will be automatically updated over time, providing valuable information to end users, business managers and service providers.

Using crowds as sensors [20] offers several advantages over using traditional physical sensors [29]. First, it provides an infrastructure-free solution, as there is no need to deploy numerous physical sensors. Second, it provides an environment-independent solution, as crowds are capable of sensing all different types of aforementioned characteristics, while physical sensors are usually limited to sense specific types of signals. Third, crowds form a much more scalable “sensor network” with a much larger coverage than traditional physical sensor networks.
Towards the goal of designing an efficient and unsupervised quantitative truth finding algorithm, we first perform a set of experiments to investigate the properties of crowdsourced quantity estimates. We find that crowds have both estimation bias and variance. By bias, we mean that the average estimation error is often not zero, and by variance, we mean that there are variations of errors around the bias. We also observe that crowds can more accurately estimate quantities in easy tasks than in difficult tasks. These findings imply that a zero-mean Gaussian distribution that is zigzag-shaped with many people. However, such task difficulties are also unknown a priori. Third, the quantitative truth needs to be found in an unsupervised manner as it is difficult to manually collect and annotate ground truth for supervised model training in reality. Fourth, existing truth finding algorithms mostly focus on aggregating conflicting categorical information (e.g., which option is most likely to be correct) and they are not directly applicable or are not efficient for aggregating noisy crowdsourced quantitative information.

To better understand the value of the proposal, we conducted an online survey to investigate the interest of the general public in knowing the quantitative characteristics of local businesses and services that we would like to provide. We posted the survey on CrowdFlower (an online crowdsourcing platform) and recruited 100 people to participate in the survey. To filter out untrustworthy responses, we inserted two test questions (about common sense), each with an obvious correct answer. Finally, 91 out of 100 participants passed the test questions, and we used their responses to generate the statistics below.

In particular, we investigate whether people are interested in knowing the real-time quantitative characteristics in the following four scenarios: 1) WaitLine: the number of people waiting in line in a cafe/restaurant/bank/shopping mall/theme park; 2) ResTab: the number of available tables in a restaurant; 3) ParkSpace: the number of available parking spaces in a parking lot; 4) GymFac: the number of available fitness machines in a gym.

Figure 2 shows the distributions of responses. It is observed that participants are more interested in information related to restaurants than that related to gyms. It makes sense as dining outside is a common activity that is performed by almost any people while going to the gym is performed by a more targeted population. Nevertheless, for all the four scenarios, more than 70% of the participants have interest in knowing their real-time status, showing the value of providing such information.

1http://www.crowdflower.com/
Figure 2: Survey results about the interest in knowing the quantitative characteristics in some real-world scenarios.

4. PROPERTIES OF CROWDSOURCED QUANTITY ESTIMATES

In this section, we investigate the properties of crowdsourced quantity estimates, which will shed light on the design of a proper human sensor model for truth finding.

In particular, we investigate the following aspects.

1. Is there any difference (e.g., in terms of perceived difficulty and estimation accuracy) if we ask people to estimate different types of quantities (e.g., count vs. percentage) with respect to the same target?

2. How accurately can people estimate quantities? Is there any bias (e.g., consistent overestimate or underestimate) and variance?

3. Does the estimation accuracy depend on the property of the target quantity itself?

4.1 Percentage Estimation is Easier than Count Estimation on Average

In the first experiment, we investigate whether there is any difference if we ask the crowds to estimate different types of quantities, e.g., count vs. percentage, with respect to the same target. This experiment is motivated by the observation that for tables in a restaurant and fitness machines in a gym, their quantities have an upper bound. For such scenarios, it seems easier to estimate a percentage instead of a count (especially for a large count), and the count can be then converted from the percentage. However, for scenarios like the number of people waiting in line which does not have an upper bound, estimating a count is the only choice.

We conducted a between-subject study on CrowdFlower, where we carefully chose 100 snapshots about ResTab, ParkSpace and GymFac (such that the percentage ranges approximately uniformly from 0 to 100 with a mean of 48.4 and a std of 26.2, while the corresponding count varies from 0 to 72 with a mean of 31.1 and a std of 16.7). We recruited 50 participants to estimate a count and another 50 participants to estimate a percentage for the target quantity in each snapshot. We then measured the difference in both participant-perceived difficulty and estimation accuracy.

We asked the participants to rate the task difficulty in three levels where 1, 2, and 3 represent “easy”, “normal” and “hard” respectively. Figure 3(a) shows the cumulative distribution functions (CDFs) of the average perceived difficulty per task. It is observed that estimating a percentage is on average easier than estimating a count for human beings, as the average perceived difficulty for 80% of the percentage estimation tasks is below 1.9 while that for 80% of the count estimation tasks is below 2.4. Figure 3(b) shows the cumulative distributions of the mean absolute error (MAE, i.e., the absolute error averaged over participants) per task, where we have converted the percentage to the corresponding count. It is observed that a majority of errors in estimating a percentage are similar to those in estimating a count, but estimating a percentage results in much fewer large errors.

According to these observations, we decide to ask crowd participants to estimate a count for line length-related tasks (e.g., WaitLine), but to estimate a percentage for occupancy-related tasks (e.g., ResTab, ParkSpace and GymFac).

4.2 Crowds have Both Bias and Variance

In the second experiment, we investigate how accurately each crowd participant can estimate counts and percentages, and whether there exist bias and variance in her estimates. We conducted experiments both on CrowdFlower and in real-world settings (details are presented in Section 7.1). In order not to make the figures cluttered, we plot the estimation errors (differences between the estimated and true values) for 24 example participants for each type of quantity estimation tasks (there is no correspondence between the participant IDs in the two tasks) in Figure 4(a)-(b).

We make three key observations. First, each participant has her own specific error pattern. Some participants have balanced errors around zero (e.g., participant 24 in Figure 4(a)), some tend to underestimate (e.g., participant 2 in Figure 4(a)), and some tend to overestimate (e.g., participant 14 in Figure 4(a)). This suggests that we need to model the ability separately for each participant instead of globally.

Second, participants’ estimates have both bias and variance. By bias, we mean that the average error is often not zero, and by variance, we mean that there are variations of errors around the bias. As a consequence, we cannot simply assume that participants’ estimates are centered around the true quantity values with zero mean and unknown variance, but have to consider both the unknown estimation bias and unknown variance.

Third, each participant may have multiple bias-variance pairs. We first take a look at Figure 4(a) for count estimation. We observe that for participant 3, a majority of errors are between −5 and 20, but a fraction of errors also largely spread out between 50 and 250. This suggests that approximating each participant’s bias and variance using a single
Gaussian distribution may not be enough. To illustrate this point, we plot the probability density functions (PDFs) by fitting the estimation errors of participant 3 using a single Gaussian distribution, a mixture of three Gaussian distributions and kernel density estimation (KDE) [2] in Figure 4(c). As KDE is non-parametric, we can consider it as a close approximation to the underlying distribution. It is observed that the PDF by fitting a single Gaussian largely deviates from that by KDE, while the PDF by fitting a Gaussian mixture model is much more similar to that by KDE. The three components of the Gaussian mixture model are centered around 3.1, 17.6 and 121.4, accounting for 55.9%, 24.7% and 19.4% of errors respectively. This suggests that using a Gaussian mixture model for each participant’s bias and variance is more appropriate. Moreover, according to [2], a Gaussian mixture model can approximate any continuous distribution to arbitrary accuracy by adjusting model parameters. It thus allows flexibility to model crowdsourced estimates with complex underlying distributions.

We note that a large percentage error does not necessarily translate to a large count error, as a percentage is normalized by the upper bound of the target quantity. For example, for a quantity with a count upper bound of 30, a percentage error of 10 is equivalent to a count error of only 3.

### 4.3 Task Difficulty Impacts Estimation Accuracy

In the third experiment, we investigate whether participants’ quantity estimation accuracy depends on the property of the quantity itself (the setup for this experiment is the same as that for the second experiment).

As we have observed that a participant may have multiple bias-variance pairs, we hypothesize that this phenomenon is caused by different difficulties in different tasks. Figure 5 shows the scatter plot of the MAEs of the crowdsourced estimates versus the true quantity values.

For count estimation (Figure 5(a)), it is observed that when the true quantity is small (e.g., below 15), the MAE is also small, meaning that most participants can accurately estimate a small count. However, when the true quantity is large (e.g., above 30), the MAE increases significantly, meaning that most participants are not good at estimating a large count. We also observe some variations. For example, the MAE when the true quantity is 20 can be larger than that when the true quantity is 30. This is because the difficulty for count estimation is also impacted by other factors such as how the line is shaped (e.g., straight or zigzag-shaped).

For percentage estimation (Figure 5(b)), we observe a different pattern. The MAEs are the largest when the true percentage is around 30 to 70, but are the smallest when the true percentage approaches an extreme value (0 or 100). This is reasonable, as it is easier to accurately estimate a percentage for a completely occupied or unoccupied space. We also observe some variations. This is because the difficulty for percentage estimation also depends on other factors such as how the occupied facilities are distributed (e.g., clustered or scattered).

Figure 5: Mean absolute error vs. true quantity value for (a) count estimation and (b) percentage estimation. (Outlier participants (e.g., providing constant estimates) are removed in order to observe a meaningful pattern.)

### 5. MODEL DESIGN

In this section, we design a probabilistic graphical model to automatically find true quantity values from noisy crowdsourced quantity estimates. We call this model the Truth, Bias and Precision (TBP) model as it models latent truth, participants’ estimation bias and precision\(^2\) (as well as task difficulties and crowdsourced estimates) based on the observations in the last section. The TBP model is a quite general framework that can be easily tuned to accommodate different types of quantity estimation tasks, and be easily extended to capture additional environmental and human factors. We list the notations used in this paper in Table 1 and illustrate the model structure in Figure 6. In the following, we detail our intuitions and the model components.

#### 5.1 Latent Truth

We model that the target quantity in each estimation task has a latent true value \(z_j\), which is generated from a Gaussian distribution.

\(^2\)Precision is the inverse of variance. We use precision instead of variance because of computational convenience.
Meaning set of quantities that hyperparameters for all the estimates on # of participants, target quantities and difficulty true value of the

We impose prior probability on each precision parameter

The precision is low.

As each estimation task may have different difficulties, we model that each participant’s estimation ability is

Moreover, it is observed from (2) that although the bias $h_{i,k}$ impacts the mean estimate value (i.e., the estimate is centered at $z_j + h_{i,k}$ instead of $z_j$); the chosen precision $\lambda_{i,k}$ impacts the precision of the estimate. As such, the impact from the true quantity value $z_j$, the task difficulty $r_{j,k}$ and the participant’s ability $h_{i,k}, \lambda_{i,k}$ are all considered and linked when modeling the generation of the crowdsourced estimate $x_{i,j}$.

In the above expression, $p(x_{i,j} = v)$ is approximated by $p(v - 0.5 \leq x_{i,j} < v + 0.5)$ when we use a Gaussian distribution to approximate $p(x_{i,j} | z_j, r_{j,k}, h_{i,k}, \lambda_{i,k})$. Moreover, we model that the difficulty level indicator $r_{j,k}$ impacts which ability parameter the participant will use (according to $k$): the hidden true quantity value $z_j$ and the chosen estimation bias $h_{i,k}$ impact the mean estimate value (i.e., the estimate is centered at $z_j + h_{i,k}$ instead of $z_j$); the chosen precision $\lambda_{i,k}$ impacts the precision of the estimate. As such, the impact from the true quantity value $z_j$, the task difficulty $r_{j,k}$ and the participant’s ability $h_{i,k}, \lambda_{i,k}$ are all considered and linked when modeling the generation of the crowdsourced estimate $x_{i,j}$.

As the task difficulty $r_{j,k}$ is not observed, we can derive the likelihood of observing $x_{i,j}$ (given model parameters but not latent variables) as

It shows that $x_{i,j}$ essentially follows a Gaussian mixture distribution (which can approximate any continuous distribution to arbitrary accuracy by adjusting model parameters [2]). Our model thus allows flexibility to model crowdsourced estimates with complex underlying distributions.

Moreover, it is observed from (2) that although the bias $h_{i,k}$ and the precision $\lambda_{i,k}$ of each mixture component are participant dependent (through $i$), the mixture coefficients $\pi_k$ are shared by all the participants and are determined by the distribution of the task difficulty. It thus allows us to link the effect of task difficulty to human estimation ability.

Besides TBP, there are also other model choices. One alternative is to model the task difficulty $r_j$ as a continuous variable, and to model only one $(h_i, \lambda_i)$ pair for each participant. We can then model the probability of observing $x_{i,j}$ as

### 5.4 Crowdsourced Estimate

Finally, we model the conditional probability that a participant $u_i$ makes an estimate $x_{i,j}$ on a quantity $z_j$ (given
behavior of random guessing to space limitation, we leave the development and evaluation will be automatically downweighted in estimating will normally result in low precisions and thus such estimates. This is because such behavior that we observe the historical median \( z_i \) makes an estimate \( x_{i,j} \) following a copying model (e.g., first pick a friend and then copy her estimate); with probability \( \pi_{i,j} \), \( x_{i,j} \) makes an estimate \( z_{i,j} \) following our TBP model. As a consequence, the probability that we observe \( x_{i,j} \) is given by

\[
p(x_{i,j}) = \pi_{i,j} p(x_{i,j}|\text{copying model}) + (1-\pi_{i,j}) p(x_{i,j}|\text{TPM model}).
\]

We can tackle lying similarly. A possible lying model could be \( p(x_{i,j}|z_j, r_{j,k}, c_{i,k}, h_{i,k}, \lambda_{i,k}) = \mathcal{N}(x_{i,j}|c_{i,k} - z_j, 1/\lambda_{i,k}) \). It means that \( u_i \) deliberately reports a reversed value \( c_{i,k} - z_j \), i.e., when \( z_j \) is large (small), \( u_i \) reports a small (large) value (the bias term \( h_{i,k} \) has been absorbed into \( c_{i,k} \)). Due to space limitation, we leave the development and evaluation of these extended models in our future work.

We note that our TBP model can naturally handle the behavior of random guessing. This is because such behavior will normally result in low precisions and thus such estimates will be automatically downweighted in estimating \( z_j \).

### 6. MODEL INFERENCE

In this section, we discuss how to perform model inference to estimate the true quantity values \( z_j \) and related model parameters in TBP. We develop two versions of model inference algorithms, one is a batch version and the other is an online version. As a consequence, the proposed model can be infrequently retrained in a batch mode on recent cumulative data (in an unsupervised manner), and then be used for online truth finding as new data arrive.

#### 6.1 Batch Inference

In our TBP model, we treat \( r_{j,k} \) as latent variables and \( \theta = \{\pi_{k}, z_j, h_{i,k}, \lambda_{i,k}\} \) as model parameters. We can write out the joint probability of crowdsourced estimates \( X = \{x_{i,j}\} \), latent variables \( R = \{r_{j,k}\} \) and model parameters \( \theta \) as

\[
p(X, R, \theta) = p(R)p(X|R, \theta)p(\theta) = \prod_j \prod_k \left[ p(r_{j,k}) \prod_{i \in U_j} p(x_{i,j}|z_j, r_{j,k}, h_{i,k}, \lambda_{i,k}) \right]^{r_{j,k}} \times \prod_j \prod_k p(z_j) \prod_i p(\lambda_{i,k}).
\]

By maximizing the logarithm of the above probability with respect to the latent variables and the model parameters, we can obtain their estimates. However, direct optimization is difficult as \( \mathbf{R} \) can take only integer values. We thus develop an efficient model inference method based on the Expectation Maximization (EM) algorithm [2, 6] and we summarize the procedure in Algorithm 1. The EM algorithm consists of an E-step and an M-step. It iterates over these two steps until model convergence.

#### 6.1.1 E-step

In the E-step, we compute the expectation of the latent variables \( r_{j,k} \) given the current estimate of model parameters \( \theta^{(t)} \), where the superscript \( t \) denotes the \( t \)th iteration. As \( r_{j,k} \) follows a multinomial distribution, we have

\[
\gamma_{j,k}^{(t)} \equiv \mathbb{E}_{R|X, \theta^{(t)}}(r_{j,k}) = p(r_{j,k}|x_j, \theta^{(t)}) \propto p(r_{j,k} | \pi^{(t)}_{j,k}) p(x_j | r_{j,k}, \theta^{(t)})
\]

\[
= p(r_{j,k} | \pi^{(t)}_{j,k}) \prod_{i \in U_j} p(x_{i,j} | r_{j,k}, z_j^{(t)}, h_{i,k}^{(t)}, \lambda_{i,k}^{(t)})
\]

\[
= \pi_{j,k}^{(t)} \prod_{i \in U_j} \mathcal{N}(x_{i,j}|z_j^{(t)}, h_{i,k}^{(t)}, 1/\lambda_{i,k}^{(t)}).
\]

As the above expression contains the product of multiple probabilities, we first compute its logarithm and then convert the value back by taking the exponential to avoid numerical underflow.

After normalization, we have

\[
\gamma_{j,k}^{(t)} = \gamma_{j,k}^{(t)} \sum_{k'} [\pi_{j,k'}^{(t)} \prod_{i \in U_j} \mathcal{N}(x_{i,j}|z_j^{(t)}, h_{i,k'}^{(t)}, 1/\lambda_{i,k'}^{(t)})]^{-1}
\]

#### 6.1.2 M-step

In the M-step, we re-estimate the model parameters \( \theta \) given the expectation \( \gamma_{j,k}^{(t)} \) of the latent variables. We perform a maximum a posteriori (MAP) estimation [2] and the corresponding M-step is to maximize the following objective

\[
\theta^{(t+1)} = \arg \max \theta \mathcal{L}(\theta) \equiv Q(\theta, \theta^{(t)}) + \log p(\theta)
\]

s.t. \( \sum_k \pi_k = 1, \)

where \( Q(\theta, \theta^{(t)}) \equiv \mathbb{E}_{R|X, \theta^{(t)}} [\log p(X, R|\theta)] \).

The solutions to this problem are given as follows (the detailed derivation is presented in Appendix A)

\[
\pi_k^* = \frac{\gamma_{j,k}^{(t)}}{N}
\]

\[
h_{i,k}^* = \frac{1}{\nu_j} \sum_{j \in U_i} \gamma_{j,k}^{(t)} (x_{i,j} - z_j)
\]

\[
\lambda_{i,k}^* = \frac{1}{\nu_j} \sum_{j \in U_i} \gamma_{j,k}^{(t)} (x_{i,j} - z_j - h_{i,k}^{(t)})^2 + h_{i,k}^{(t)}
\]

\[
z_j^* = \frac{\nu_j \mu_j + \sum_k (\gamma_{j,k}^{(t)} \sum_{i \in U_j} \lambda_{i,k}^{(t)} - h_{i,k}^{(t)})}{\nu_j + \sum_k (\gamma_{j,k}^{(t)} \sum_{i \in U_j} \lambda_{i,k}^{(t)})}
\]

where \( U_j \) is the set of participants who make a quantity estimate on \( z_j \).
As a consequence, we initialize data to form a new estimates setting in Appendix B. We present the model initialization and hyperparameter solution for 

\[
\pi^* = \arg \max_i \pi_i \quad \text{s.t.} \quad \sum_i \pi_i = 1
\]

and it is 0 otherwise.

where \( Z_i \) is the set of quantities that \( u_i \) makes an estimate on.

We observe that except \( \pi_i^* \), we cannot obtain a closed-form solution for \( h_i^*, \lambda_i^* \) and \( z_i^* \) as they are coupled together. As a consequence, we initialize \( z_i \) and then iterate over (7)-(9) until convergence in each M-step to obtain \( h_i^*, \lambda_i^* \) and \( z_i^* \). We present the model initialization and hyperparameter setting in Appendix B.

### 6.2 Online Inference

The above batch inference algorithm requires a matrix \( X \) of crowdsourced quantity estimates involving multiple participants and multiple target quantities as the input. In an online setting, when crowdsourced estimates \( x_o \) are made on a new quantity \( z_o \), we need to augment \( x_o \) with historical data to form a new \( X \) and rerun the EM algorithm, which is costly. We thus design an online inference algorithm below which can work on \( x_o \) directly.

In the online inference, we compute the posterior probability of the true quantity value \( z_o \) given current crowdsourced estimates \( x_o \) as

\[
p(z_o|x_o) = \frac{p(z_o)p(x_o|z_o)}{p(x_o)} = \frac{p(z_o) \sum_k p(r_o,k)p(x_o|z_o,r_o,k)}{p(x_o)}
\]

\[
= \sum_k \pi_k \mathcal{N}(z_o|x_o,1) \prod_i \mathcal{N}(x_i,o|z_o+h_i,k,1/\lambda_i,k)
\]

Where \( \mathcal{N}(\cdot) \) is the normal distribution.

We compute a minimum mean square error (MMSE) estimation [2] for \( z_o \), which is given by

\[
E(z_o|x_o) = \sum_k \pi_k \frac{\mu_o + \sum_{i \in U_o} \hat{h}_{i,k} (x_{i,o} - \hat{h}_{i,k})}{\nu_o + \sum_{i \in U_o} \lambda_{i,k}}
\]

\[
= \sum_k \pi_k \frac{\hat{x}_{o,k}^T \lambda_{o,k}^{-1} \hat{x}_{o,k}}{1^T \lambda_{o,k}}
\]

(10)

where

\[
\hat{x}_{o,k} = [\mu_o, x_{1,o} - \hat{h}_{1,k}, x_{2,o} - \hat{h}_{2,k}, \ldots]^T,
\]

\[
\hat{\lambda}_{o,k} = [\nu_o, I_{1,o} \hat{\lambda}_{1,k}, I_{2,o} \hat{\lambda}_{2,k}, \ldots]^T.
\]

\( I_{i,o} = 1 \) if \( i \in U_o \) and it is 0 otherwise.

The expression (10) follows from the technique of “completing the square” for Gaussian distributions [2]. We observe that \( E(z_o|x_o) \) can be simply expressed in terms of the inner product of vectors and thus can be computed fast and efficiently. Finally, we round \( E(z_o|x_o) \) to the closest integer as the estimate \( \hat{z}_o \).

Taking a close inspection on (10) reveals that the effective estimate by \( u_i \) is actually \( x_{i,o} - \hat{h}_{i,k} \) for the difficulty level \( k \) where the bias should be removed. These effective estimates are then weighted by \( \frac{1}{1^T \hat{\lambda}_{o,k}} \) where a large \( \hat{\lambda}_{i,k} \) implies a more reliable participant. For difficulty level \( k \), we have an estimate \( \hat{z}_{o,k} \equiv \frac{\hat{x}_{o,k}^T \hat{\lambda}_{o,k}^{-1} \hat{x}_{o,k}}{1^T \hat{\lambda}_{o,k}} \) which is the weighted sum of the effective estimates. Each \( \hat{z}_{o,k} \) is then weighted by \( \pi_k \) and combined to form the final estimate \( \hat{z}_o \). We can also compute the variance of \( \hat{z}_o \), which can be used as the confidence of the estimate.

## 7. EXPERIMENTS

In this section, we present our experimental results to demonstrate the effectiveness of TBP compared with other state-of-the-art algorithms.

### 7.1 Setup

We conducted two sets of experiments.

**Crowdsourcing platform-based studies:** We conducted the first set of experiments over an online crowdsourcing platform. We searched for snapshots of scenarios corresponding to WaitLine, ResTab, ParkSpace and GymFac through Google Images. We then masked the human faces in these snapshots and posted them on CrowdFlower for crowdsourced quantity estimation (WaitLine is used for count estimation while others are used for percentage estimation). We annotated ground truth by careful counting and converting count to percentage when necessary. Such an experimental setup allows us to examine the human ability in quantity estimation in a wide range of scenarios, e.g., in restaurants, banks, shopping malls, theme parks, parking lots, and gyms. Moreover, it allows us to easily recruit a large number of participants and to make fair comparisons between different groups of participants on estimating the same target quantities (e.g., for the observations in Section 4.1). We finally recruited 106 participants for 400 count estimation tasks and 93 participants for 400 percentage estimation tasks.

**Case studies:** To evaluate our model in real-world settings, we conducted a second set of experiments\(^3\). It consists of two case studies, one for estimating waiting line length

\[^3\]These experiments have been approved by IRB.

### Table 2: Statistics of crowdsourced quantity estimates (pt - participant, est - estimate, avg - average, case - case study, cf - CrowdFlower).

<table>
<thead>
<tr>
<th>Task</th>
<th># tasks</th>
<th># pts</th>
<th>avg # per task</th>
<th># avg # tasks per pt</th>
<th># total ests</th>
<th>ground truth quantity (mean ± std)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count (cf)</td>
<td>400</td>
<td>106</td>
<td>20</td>
<td>75.5</td>
<td>8,000</td>
<td>29.4 ± 16.1</td>
</tr>
<tr>
<td>Percent (cf)</td>
<td>400</td>
<td>93</td>
<td>20</td>
<td>86.0</td>
<td>8,000</td>
<td>46.9 ± 28.3</td>
</tr>
<tr>
<td>Count (case)</td>
<td>327</td>
<td>22</td>
<td>3.5</td>
<td>51.4</td>
<td>1,131</td>
<td>22.5 ± 9.4</td>
</tr>
<tr>
<td>Percent (case)</td>
<td>282</td>
<td>20</td>
<td>3.0</td>
<td>42.3</td>
<td>845</td>
<td>56.7 ± 20.4</td>
</tr>
</tbody>
</table>
(count estimation) in a restaurant and the other for estimating occupancy level of seats (percentage estimation) in a room of a library. We recruited participants who reported to frequently visit the restaurant during [11am, 1pm] or the library during [1pm, 4pm]. Finally, 22 participants were recruited for the count estimation task and 20 for the percentage estimation task. The experiments were conducted over two weeks. We collected ground truth in the desired time interval. Each crowd participant made quantity estimates whenever she would like to through an app (which limited an estimate per minute for the same place). We then considered the estimates made within the same one minute interval and three minute interval as corresponding to the same task for the restaurant and the library respectively. Retaining tasks with at least two crowdsourced estimates, we finally had 327 count estimation tasks and 282 percentage estimation tasks.

The statistics of these experiments are listed in Table 2. In the following, we present experimental results on the CrowdFlower-based studies (“cf”) in Sections 7.4–7.6 and those on the real-world case studies (“case”) in Section 7.7.

7.2 Methods in Comparison

We compare the performance of the following algorithms in recovering the true quantity values from noisy crowdsourced estimates. 1) **MV**: the widely used majority voting method; in quantitative truth finding, it is equivalent to finding the mode in the data. 2) **AvgLog**: the AverageLog method proposed in [15]. 3) **Invest**: the Investment method proposed in [15]. 4) **TF**: the Truth Finder method proposed in [30] which contains a mechanism to encode implication scores between similar observations. We set the implication score between $x_i,j$ and $x_{i',j'}$ as $\exp(-|x_i,j - x_{i',j}|/20)$, which increases as the difference between $x_i,j$ and $x_{i',j}$ decreases. 5) **Median**: as our truth finding problem is quantitative, it makes more sense to use the median (more robust to outliers than mean) of crowdsourced values as the estimate. 6) **TBP**: our proposed truth, bias and precision model (the detailed setting is provided in Appendix B). The last two methods round their output to the closest integer.

Among the above methods, the first four are mainly designed for categorical truth finding. They aim to find the best possible truth from a mutually exclusive candidate set, which is $C_j = \{x_{i,j} | i \in U_j\}$ in our problem. For example, if for the $j$th target quantity, $x_{1,j} = 10$, $x_{2,j} = 8$, $x_{3,j} = 10$ and $x_{4,j} = 20$, we then have $C_j = \{8, 10, 20\}$. In contrast, the last two methods may output a value that is not in $C_j$ as they are more tailored to quantitative truth finding. In case no participant makes an accurate estimate, these two methods may still be able to output the correct truth.

Although there are other truth finding methods proposed for crowdsourced image classification [17], social sensing [21] and crowdsourced detection of spatial events [12], they are mainly designed for binary truth and extending them to multinomial truth and quantitative truth is problematic. We thus do not include them for comparison.

7.3 Evaluation Metric

We use the root mean square error (RMSE) to evaluate the accuracy of truth finding methods. It is defined as

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_j (\hat{z}_j - z_j)^2}.$$
Figure 9: (a) Convergence rate of TBP. (b) Average CPU time vs. $N$ when $M = 20$ for count estimation (cf). (c) Average CPU time vs. $M$ when $N = 200$ for count estimation (cf).

Figure 10: RMSE and CPU time vs. # of crowdsourced estimates for online truth finding with $N = 1$ (cf). (a) RMSE of count estimation. (b) RMSE of percentage estimation. (c) CPU time of count estimation.

**Percentage estimation:** Figure 8 plots the RMSEs in crowdsourced percentage estimation (cf). We observe similar trends as those in Figure 7. $\text{AvgLog}$, $\text{Invest}$ and $\text{TF}$ perform worse than $\text{Median}$, and $\text{TBP}$ still performs best.

### 7.5 Computation Time

Figure 9(a) shows that $\text{TBP}$ typically converges in only a few iterations. Figure 9(b)-(c) show the computation times of different methods for count estimation (those for percentage estimation are similar or even shorter and thus are not shown). It is observed that the computation time generally increases as more crowdsourced estimates are involved. $\text{TBP}$ is the most time-consuming, followed by $\text{TF}$. However, even when there are $M = 20$ participants and $N = 200$ target quantities, $\text{TBP}$ only takes on average 0.6 second to converge, showing that it is computationally fast and efficient even in a batch mode.

### 7.6 Online Truth Finding

In a real-world setting, it is important for an algorithm to perform online truth finding from only a few (e.g., 2-5) crowdsourced quantity estimates collected over a short time window. We thus examine the performance of the online version of $\text{TBP}$. To implement an online version of $\text{AvgLog}$, $\text{Invest}$ and $\text{TF}$, we take a similar approach that we use for $\text{TBP}$. That is, we assume the source trustworthiness have converged in respective batch inference and we then use these converged values to compute the confidence score for each possible truth value online.

We use 10-fold cross validation where in each fold 90% of data are used to train the model in a batch mode (unsupervised) and 10% are used for online truth finding. Figure 10 depicts the RMSEs and CPU times when the number of online crowdsourced estimates ranges from 2 to 8. It is observed that $\text{TBP}$ performs much better than the other methods when very few crowdsourced estimates are available. For count estimation, when there are only two crowdsourced estimates, $\text{TBP}$ exhibits an error reduction (in terms of RMSE) of 20.0% and 26.4% with respect to strong competitors $\text{Median}$ and $\text{TF}$. Similarly, for percentage estimation, when there are only two crowdsourced estimates, $\text{TBP}$ exhibits an error reduction of 22.2% and 25.0% with respect to $\text{Median}$ and $\text{TF}$. Moreover, the paired-sample t-tests show that the differences in RMSEs by $\text{TBP}$ and other methods are statistically significant in all cases (with $p < 10^{-4}$). These results demonstrate that $\text{TBP}$ is able to accurately recover the truth according to Equation (10) when very few participants make observations, and thus is extremely useful for online truth finding.

In terms of the computation time, it is observed from Figure 10(c) that all the methods can perform online truth finding for a single target quantity ($N = 1$) within 0.12 millisecond, which are extremely computationally light.

### 7.7 Case Studies

Figure 11 and Figure 12 plot the results of a case study on estimating the waiting line length in a restaurant and a case study on estimating the occupancy level of seats in a library respectively.

Similar to the CrowdFlower-based studies, we observe that the RMSEs are relatively stable with respect to the changing number of target quantities $N$ given a fixed number of participants $M$. However, we also observe that the extent that the RMSEs decrease with the increasing number of participants is not as significant as that in the CrowdFlower-based studies. Some methods (e.g., $\text{MV}$, $\text{AvgLog}$, $\text{Invest}$ and $\text{TF}$ in Figure 11(b)) even show relatively stable RMSEs when more participants are present.

This may be because the crowdsourced estimate matrix $X$ is much sparser in these case studies than in the CrowdFlower-
8. DISCUSSION

Predicting future characteristics. Besides recovering instantaneous characteristics, forecasting future characteristics is also an important and useful functionality. One possibility is to first recover instantaneous characteristics and then build a prediction model on these recovered data. The accuracy of our TBP model also facilitates accurate prediction. An alternative is to extend our TBP model to jointly model historical crowdsourced estimates and future characteristics.

Exploiting physical constraints. Physical constraints can help us filter out untrustworthy data before employing our TBP model. One physical constraint that we can leverage is the participant’s location [12, 22]. If a participant is not physically present at a desired location when making estimates, her information credibility is low. We can thus leverage localization or location clustering techniques to filter out untrustworthy data.

Size of crowd. Statistical models need sufficient data to work well and thus our TBP model may fail if the size of crowd is small. It is thus useful to investigate the Cramer-Rao lower bound (CRLB) [22] and to examine how the lower bound on the estimation error changes with respect to different sizes of crowd.

Alternative error models. We currently consider the crowdsourced estimation error as additive to the true quantity. An alternative is to examine the ratio between the estimated and the true quantity, and explore a multiplicative error model. Once we replace the crowdsourced data and estimated and the true quantity, and explore a multiplicative model. Once we replace the crowdsourced data and the true quantities by their logarithms, our TBP model can be applied again to model this multiplicative relationship.

Other application scenarios. Besides in businesses and services, the ability to accurately recover quantitative truth from noisy crowd sensors would also be useful in other scenarios, such as tallying people gathering at public places for early event detection, and tallying casualties of a natural disaster for efficient aid. We also note that some businesses have queue ticket systems that provide ticket numbers and even queue length (easy to share online). In such cases, there is clearly no need to use crowds as sensors. However, our approach remains valuable for many other scenarios where such systems are not available or not applicable.

9. RELATED WORK

Crowds as sensors. Research in crowdsourcing has gained rapid growth in recent years [16]. Crowds have been explored to perform various Human Intelligence Tasks (HITs) such as large-scale image classification [17], transcription [9] and word processing [1]. Besides performing such HITs on crowdsourcing platforms such as Amazon Mechanical Turk and CrowdFlower, crowds have also been utilized as sensor carriers and operators [3, 8], such as to sense the pulse of a city [4] and to sense potholes on streets [10]. A recent sur-
vey on human-centric sensing [20] emphasizes the concept that humans can also serve as sensors themselves to detect and report events in the physical world, e.g., to detect earth quakes [19], to detect desired flora on campus [18] and to detect graffiti in public places [11].

In this paper, we propose a new application that is to leverage crowds as sensors to sense quantitative characteristics in local businesses and services. Such a capability has the potential to enable well informed decision for end users and timely crowd management for business managers and service providers.

Although physical sensor networks can also be deployed as an alternative solution (e.g., sound-based occupancy level inference [25] and WiFi-based waiting line characterization [26]), dedicated sensors, features and information processing algorithms need to be deployed and developed for each specific scenario. In contrast, using crowds as sensors provides a unified, infrastructure-free, environment-independent and scalable solution. However, it also poses a challenge on information credibility which we discuss below.

Truth finding. Truth finding is to derive the most accurate and integrated information from diverse and sometimes conflicting information sources [30,31].

In the domain of truth finding from conflicting Web information, Yin et al. [30] proposed truthfinder, which is a transitive voting algorithm with rules specifying how votes iteratively flow from sources to claims and then back to sources. It has been shown to be superior than majority voting and the hubs and authorities algorithm [7] which was initially designed to find popular web pages. Pasternack and Roth [15] proposed AverageLog, Investment and PooledInvestment algorithms, where sources invest their credibility in the claims they make, and claim belief is then non-linearly grown and apportioned back to the sources. Zhao et al. [31] proposed a more principled probabilistic approach which can automatically infer true claims and two-sided source quality.

In the domain of aggregating conflicting responses in crowdsourcing tasks, Dawid and Skene [5] modeled the generative process of the responses by introducing worker ability parameters. Whitehill et al. [28] further included the difficulty of the task in the model. Welinder et al. [27] proposed a model consisting of worker compatibility for each task. Wang et al. [21] proposed an algorithm that models both the truth of tasks and the reliability of workers for social sensing. They further extended their model to consider potential dependence information sources in [23].

Nevertheless, these algorithms are mainly designed for categorical truth finding and some are even dedicated for binary truth. As a consequence, these algorithms do not directly apply or are not efficient for the quantitative truth finding problem addressed in this paper. We propose a more tailored, unsupervised TBP model which consists of quantitative truth, task difficulties, and participants’ biases and precisions. In addition to designing a batch model inference algorithm, we also devise an even faster online version. As a consequence, the proposed model can be infrequently retrained in a batch mode on recent cumulative data, and then be used for online truth finding as new data arrive.

10. CONCLUSION

In this paper, we investigate the potential of leveraging crowds as sensors to sense quantitative characteristics in real-world scenarios, and investigate how to recover the true quantity values from noisy crowdsourced information. Reliably recovering such information has the potential to enable timely informed decision, crowd management, and event detection. We find that crowd sensors have both bias and precision in quantity sensing, and task difficulties impact the sensing accuracy. Based on these findings, we propose an unsupervised, probabilistic model (the TBP model) to automatically find true quantity values from noisy crowdsourced estimates. We also develop efficient batch and online model inference algorithms. Experimental results demonstrate the effectiveness of our proposed model in various scenarios.

Acknowledgements

This research is based upon work supported in part by the U.S. ARL and U.K. Ministry of Defense under Agreement Number W911NF-06-3-0001, and by the NSF under award CNS-1213140. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views or represent the official policies of the NSF, the U.S. ARL, the U.S. Government, the U.K. Ministry of Defense or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

11. REFERENCES

APPENDIX

A. DERIVATION OF SOLUTIONS IN THE M-STEP

The constrained optimization problem in (5) can be transformed to an unconstrained problem by using a Lagrange multiplier $\eta$ as

$$\theta^{(t+1)} = \arg \max_\theta f(\theta, \theta^{(t)}) + \eta \left( \sum_k \pi_k - 1 \right).$$  \hspace{1cm} (11)

The Q function in (5) is defined as $Q(\theta, \theta^{(t)}) \equiv \mathbb{E}_{R|X, \theta^{(t)}}[\log p(X, R|\theta)]$. We can then write out the objective function in (5) as

$$f(\theta, \theta^{(t)}) = \mathbb{E}_{R|X, \theta^{(t)}}[\log p(X, R|\theta)] + \log p(\pi) + \log p(\lambda)$$

$$= \sum_k \sum_j \gamma_{j,k}^{(t)} \left[ \log \pi_k + \sum_i \left( \frac{1}{2} \log \lambda_{i,k} - \frac{1}{2} \lambda_{i,k}(x_{i,j} - z_j - h_{i,k})^2 \right) \right]$$

$$+ \sum_j \left( \frac{1}{2} \log \nu_j - \frac{1}{2} \nu_j(z_j - \mu_j)^2 \right)$$

$$+ \sum_k \sum_i \left( a_{i,k} - 1 \right) \log \lambda_{i,k} - b_{i,k} = 0.$$

where $\gamma_{j,k}^{(t)} \equiv \mathbb{E}_{R|X, \theta^{(t)}}[\rho_{j,k}]$ is given by (4).

To obtain the estimates of model parameters, we set the derivative of $f(\theta, \theta^{(t)})$ with respect to $\theta$ to 0 and we have

$$\frac{\partial f}{\partial \pi_k} = \sum_j \gamma_{j,k}^{(t)} \frac{1}{\pi_k} + \eta = 0$$

$$\frac{\partial f}{\partial \nu_j} = \sum_k \gamma_{j,k}^{(t)} \sum_i \left( \lambda_{i,k} (z_j + h_{i,k} - x_{i,j}) \right) + \nu_j (z_j - \mu_j) = 0$$

$$\frac{\partial f}{\partial h_{i,k}} = \sum_j \gamma_{j,k}^{(t)} \sum_i \left( \lambda_{i,k} (h_{i,k} + z_j - x_{i,j}) \right) = 0$$

$$\frac{\partial f}{\partial \lambda_{i,k}} = \sum_j \gamma_{j,k}^{(t)} \left( \frac{1}{2 \lambda_{i,k}} - \frac{1}{2} (x_{i,j} - z_j - h_{i,k})^2 \right)$$

$$+ (a_{i,k} - 1) \frac{1}{\lambda_{i,k}} - b_{i,k} = 0.$$

Solving the above equations, we have the solutions in (6)-(9).

B. MODEL INITIALIZATION AND HYPER-PARAMETER SETTING

We initialize the E-step with

$$z_j = \text{median}_{i \in U_j} (x_{i,j}),$$

$$h_{i,k} = \frac{1}{|\mathcal{Z}_{i,k}|} \sum_{j, \hat{z}_j \in \mathcal{Z}_{i,k}} (x_{i,j} - \hat{z}_j),$$

$$\lambda_{i,k} = \frac{1}{|\mathcal{Z}_{i,k}|} \sum_{j, \hat{z}_j \in \mathcal{Z}_{i,k}} (x_{i,j} - \hat{z}_j - h_{i,k})^2,$$

where $\mathcal{Z}_{i,k}$ is the set of quantities that are in difficulty level $k$ and that $u_i$ makes an estimation on.

We set $\mathcal{L}_1 = \{z \leq 20\}$, $\mathcal{L}_2 = \{20 < z \leq 50\}$, $\mathcal{L}_3 = \{z \geq 50\}$ for count estimation with $K = 3$ and set $\mathcal{L}_1 = \{z \leq 10\}$, $\mathcal{L}_2 = \{10 < z < 90\}$ for percentage estimation with $K = 2$.

We set the hyperparameters as

$$\mu_j = \bar{z}_j, \quad \nu_j = \frac{|U_j|^2}{\sum_{i \in U_j} (x_{i,j} - \bar{z}_j)^2},$$

$$a_{i,k} = \frac{|\mathcal{Z}_{i,k}|}{2} + 1, \quad b_{i,k} = \frac{a_{i,k}}{\lambda_{i,k}}.$$