Crowdsourced POI Labelling: Location-Aware Result Inference and Task Assignment

Huiqi Hu[†], Yudian Zheng[*], Zhifeng Bao[‡], Guoliang Li[†], Jianhua Feng[†], Reynold Cheng[*]

[†]Department of Computer Science, Tsinghua National Laboratory for Information Science and Technology (TNList),
Tsinghua University, Beijing, China

[‡]Computer Science and Information Technology, RMIT University, Melbourne, Australia

[*]Department of Computer Science, The University of Hong Kong, Hong Kong, China

hhq11@mails.tsinghua.edu.cn; {liguoliang,fengjh}@tsinghua.edu.cn; ydzheng2@cs.hku.hk;zhifeng.bao@rmit.edu.au

***Abstract*—Identifying the labels of points of interest (POIs), aka POI labelling, provides significant benefits in location-based services. However, the quality of raw labels manually added by users or generated by artificial algorithms cannot be guaranteed. Such low-quality labels decrease the usability and result in bad user experiences. In this paper, by observing that crowdsourcing is a best-fit for computer-hard tasks, we leverage crowdsourcing to improve the quality of POI labelling. To our best knowledge, this is the first work on crowdsourced POI labelling tasks. In particular, there are two sub-problems: (1) how to infer the correct labels for each POI based on workers' answers, and (2) how to effectively assign proper tasks to workers in order to make more accurate inference for next available workers. To address these two problems, we propose a framework consisting of an inference model and an online task assigner. The inference model measures the quality of a worker on a POI by elaborately exploiting (i) worker's inherent quality, (ii) the spatial distance between the worker and the POI, and (iii) the POI influence, which can provide reliable inference results once a worker submits an answer. As workers are dynamically coming, the online task assigner judiciously assigns proper tasks to them so as to benefit the inference. The inference model and task assigner work alternately to continuously improve the overall quality. We conduct extensive experiments on a real crowdsourcing platform, and the results on two real datasets show that our method significantly outperforms state-of-the-art approaches.**

I. Introduction

With the popularity of location-based services, labels are generated in order to provide concise yet precise descriptions for each point of interest (POI). Previous studies have shown that searching resources based on their associated labels leads to more effective and accurate resource retrieval for users [1]. Moreover, accurate labels can also benefit other applications, e.g., activity recommendation to users [25].

However, the quality of POI labels cannot be guaranteed in reality, because anonymously incredible or malicious users may abuse the right of manual labelling, while labels automatically generated by some artificial algorithms [9,19] still involve low-quality labels due to limited accuracy of those algorithms. Therefore, it calls for an effective method to generate high-quality labels. Fortunately, crowdsourcing emerges and becomes an effective way to handle computer-hard tasks, which are difficult for computers (e.g., POI labelling). It inspires us to exploit crowdsourcing to improve the labelling quality. However, crowdsourcing is not free (as we need to pay the workers who label the POIs). To reduce the monetary cost, we can first utilize existing techniques to generate candidate labels for POIs and then ask crowdsourced workers to select correct labels from the candidate labels to ensure the quality.

In this paper we study the POI labelling problem: given a set of POIs, each of which has several candidate labels, and a budget B, we identify the correct labels for the POIs by asking at most B tasks, where each task asks workers to select correct labels from the candidate labels of a POI. In particular, there are two sub-problems to address: (1) Label Inference: how to infer the correct labels for each task based on workers' answers; (2) Task Assignment: when workers are requesting tasks, how to assign proper tasks to these workers to make more accurate inference. To our best knowledge, this is the first study on crowdsourced POI labelling.

Although many studies have investigated the answer inference problem and task assignment problem, they focus on choosing labels on objects such as images and entities [7,12,15,16,22,24,27] which do not involve the locations of tasks or workers. Actually the distance between workers and POIs has a significant impact on the label inference (see Section V for detailed justifications). Recently, spatial crowdsourcing tasks have also raised increasing attentions from the research community [4,13,14,20,21]. However, they have two main differences from our problem. First, they require workers to travel to the specific locations to answer the tasks, e.g., taking photos of a restaurant or reporting the congestion of a place; while we drop out this requirement as workers can be familiar with the POIs even when they are not at the locations at present. Second, they focus on minimizing the travel distances of workers. In contrast, we aim to improve the labelling quality.

Crowdsourced POI labelling has many challenges. First, there exist more complicated factors that can affect the answer quality for POI labelling tasks as compared to simple labelling tasks: (1) famous POIs often receive higher quality answers than ordinary ones and (2) the distance between a worker and a POI has effect on the quality and the impact varies for different workers. It is non-trivial to form these factors into one effective model to measure workers' quality and provide reliable inference results at the same time. Second, as workers are dynamically coming, it is hard to instantly

978-1-5090-2020-1/16/$31.00 © 2016 IEEE

identify workers' characteristics and judiciously assign proper tasks to them to further improve the inference quality by well exploiting the previous answers of these workers.

To address these challenges, we propose a POI-Labelling Framework (as illustrated in Figure 1) with two main components: (1) *An inference model*, which takes POI tasks and workers' answers as input and returns the inference results for each task. We develop a graphical probability inference model by elaborately exploiting (i) the worker's inherent quality, (ii) the spatial distance between the worker and the POI, and (iii) the POI influence (see Section III for details). (2) *An online task assigner*, which takes the estimated worker quality and the POI influence as input, and assigns a group of tasks for each available worker by maximizing the accuracy improvement (see Section IV). Given a cost budget (i.e., the number of allowed assignment), the inference model and the task assigner work alternately for a dynamic scenario: when workers are coming for tasks, the task assigner proceeds to generate the best-fit tasks to each worker. Then the answers are collected and the inference model proceeds to estimate the worker quality based on current answers. The measured qualities are then used by the assigner in judiciously assigning the best tasks to next round of coming workers, such that the overall accuracy of the inference results can keep growing. Such alternate process continues until the budget runs out.

To summarize, we make the following contributions.

(1) We formalize the crowdsourced POI labelling problem and its two sub-problems: label inference and task assignment (Section II).

(2) We develop an effective inference model which utilizes the spatial location information of workers and POIs to measure the worker quality and the POI influence in a finer-granularity, and utilize them to infer results (Section III).

(3) Based on the inference model, we propose an adaptive task assignment algorithm to further improve the inference accuracy (Section IV).

(4) We conduct extensive experiments on a real crowdsourcing platform, and the results show that our methods significantly outperforms state-of-the-art approaches (Section V).

II. Problem Statement

POI Labelling Problem. Given a set of POI labelling tasks $T = \{t_1, t_2, \ldots, t_{|T|}\}$, each task $t = \{O_t, L_t\}$ includes a POI O_t (with a name and a geo-location) and a label set $L_t = \{l_{t,1}, l_{t,2}, \ldots, l_{t,|L_t|}\}$. For simplicity of presentation, in the rest of the paper we use task t and POI O_t interchangeably and assume that each task has the same number of labels unless specified otherwise. But note that our method can support the case that different tasks have different number of labels. Each label $l_{t,i}$ ($1 \le i \le |L_t|$) has a *true result* 1/0 ("yes/no"), where 1 (0) indicates that $l_{t,i}$ is *a correct (an incorrect)* label for O_t.

Workers. Each worker w has a location (e.g. home, office). For each task $t = \{O_t, L_t\}$, workers are asked to select labels from L_t which they think correct for O_t. We denote the answer set by $\mathcal{R} = \{(w, t, \mathcal{R}(w,t))\}$, where $\mathcal{R}(w,t) = \{r_{w,t,k} \mid 1 \le k \le |L_t|\}$ is w's answer for a task t and $r_{w,t,k} = 1/0$ is w's answer for a certain label $l_{t,k}$. Figure 2 shows a labelling task for the POI "Beijing Olympic Forest Park". Among the ten labels, if a worker w thinks "1. Park" is a correct label for this POI by ticking its box, then $r_{w,t,1} = 1$; otherwise $r_{w,t,1} = 0$.

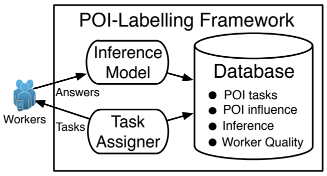
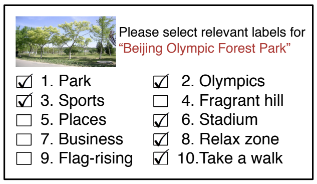


Fig. 1. POI Labelling Framework



Please select relevant labels for "Beijing Olympic Forest Park"

☑ 1. Park ☑ 2. Olympics
☑ 3. Sports □ 4. Fragrant hill
□ 5. Places ☑ 6. Stadium
□ 7. Business ☑ 8. Relax zone
□ 9. Flag-rising ☑ 10.Take a walk

Fig. 2. An Example of Task

Task Assignment. When a set W of available workers are requesting for tasks, it needs to assign h tasks (i.e., a human-intelligence task) to each worker. We denote the tasks assigned to W as $\mathcal{A}(W) = \{\mathcal{A}(w) | w \in W\}$ where $\mathcal{A}(w)$ is the set of h tasks assigned to worker w. A worker can do multiple tasks and we denote $T(w) = \{t \in T | t \text{ is done by } w\}$ as the set of tasks already done by worker w. Meanwhile, a task can be answered by several workers, we denote $W(t) = \{w | w \text{ has done task } t\}$ as the set of workers who have done task t.

Problem Description. The crowdsourced POI labelling problem aims to deduce the correct labels for each POI. We use accuracy to evaluate the crowdsourced framework, which is the average percentage of accurately deduced[1] labels (returned by an algorithm) among all labelling tasks, i.e.

$$\text{accuracy} = \frac{1}{|T|} \cdot \sum_{t_i \in T} \frac{N_{t_i}}{|L_{t_i}|}, \tag{1}$$

where N_{t_i} is the number of labels that an algorithm accurately reports for task t_i. For example, assuming $|L_{t_i}|$=10 for task t_i and the first 3 labels are the true labels; if an algorithm identifies the 1^{st} and 4^{th} label as the correct ones, then N_{t_i}= 7.

There are two sub-problems to study in achieving a high-quality crowdsourced POI labelling. (1) The result inference problem: given the answer set $\mathcal{R}$ from workers, how to infer the correct labels for each POI? (2) The task assignment problem: when a set W of available workers request tasks, how to assign h proper tasks to each worker? Since we cannot predict online workers in future and optimize the overall accuracy at once, we alternately maximize the accuracy improvement for the current workers W. Thus we can achieve an optimized accuracy step by step until the given budget runs out. Next we give the formal definition.

Definition 1 (Crowdsourced POI Labelling): Given a set of tasks T and a budget B, the Crowdsourced POI Labelling repeats the following two steps:
(1) Label Inference: when workers submit answers, infer the true labels for all tasks based on workers' answer set $\mathcal{R}$;
(2) Task Assignment: when workers request tasks, find an optimal assignment $\mathcal{A}(W)$ to maximize the improvement of overall accuracy, if the budget does not run out.

III. Inference Model

In this section, we propose an inference model to infer the labels of POIs given the current answer set $\mathcal{R}$ returned from workers. We first introduce our intuitions, then describe the details of the model, and finally discuss how to compute the parameters in our model.

[1] We consider both *correct* and *incorrect* labels in computing accuracy.

A. Model Overview

(1) **Worker Quality**. It includes two parts. (i) Worker's *Inherent Quality*. Workers have diverse quality due to their ability and background knowledge. The workers with low inherent quality, such as spammers and workers without any knowledge about the POIs, are error-prone to answer the tasks. (ii) *Distance-aware Quality*. The quality of a worker on a POI is also influenced by the distance between the POI and the worker. Intuitively, a worker can give more accurate answers to nearby tasks than distant tasks, as workers are usually more familiar with nearby POIs. Apparently, this influence of distance varies for different workers. In general, some workers only have good knowledge for a few POIs, so they can give accurate answers only for nearby POIs. On the contrary, some workers may be less sensitive to the distance, and they may provide accurate answers even if the distance is large.

(2) **POI-Influence.** We also need to consider the influence of a POI that can affect the labelling quality. On the one hand, some POIs are famous and have large influences, and they are easy to receive correct answers as most workers have background knowledge on them. On the other hand, some POIs have small influences, and only nearby workers may know the POIs. For example, Beijing Olympic Park should have a larger influence than Beijing Botanical Park.

B. Model Details

To model the quality of a worker and the influence of POIs, we propose a probability model. In the probability model, both the worker quality and the POI-influence are modeled by parameters of random variables. We first describe the model and then introduce how to estimate these parameters in Section III-C.

Result Modeling. Since the ground truth of a label is unknown, we use a binary random variable $z_{t,k}$ to represent the true result of label $l_{t,k}$. If $l_{t,k}$ is a correct label of a POI O_t, $z_{t,k} = 1$; otherwise $z_{t,k} = 0$. $z_{t,k}$ satisfies a Bernoulli distribution where $P(z_{t,k} = 1)$ denotes the probability that $l_{t,k}$ is a correct label. We use $P(z_{t,k})$ to infer the true result for $l_{t,k}$, and if $P(z_{t,k} = 1) \geq 0.5$, we infer $l_{t,k}$ as a correct label of task t.

Worker's Inherent Quality. We use a random variable i_w to represent the inherent quality of worker w, which is a binary variable: $i_w = 1$ if w is a well-qualified worker; $i_w = 0$ if w is an unqualified worker (e.g. a spammer, an irresponsible worker, or worker without good knowledge on all POIs). i_w satisfies the Bernoulli distribution, i.e., $P(i_w = 1)$ and $P(i_w = 0) = 1 - P(i_w = 1)$ represent the probability that w is a qualified and an unqualified worker respectively.

Definition 2 (Worker's Inherent Quality): We define the inherent quality of a worker w as

$$\mathtt{WQ}_w = P(i_w). \tag{2}$$

A higher $\mathtt{WQ}_w$ derives a better inherent quality of worker w.

Distance-Aware Quality. Let $d(w,t)$ denote the normalized euclidean distance between a worker w and a task t ($0 \leq d(w,t) \leq 1$). [2]. A smaller $d(w,t)$ derives a larger probability that w correctly answers t. Any function satisfying this property can be used to compute the distance-based quality. Here we take the bell-shaped function as an example and our techniques are applicable to any other functions.

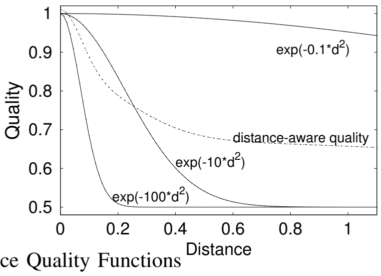


Fig. 4. Distance Quality Functions

Definition 3 (Bell-Shaped Function):

$$f_\lambda(d(w,t)) = \frac{1 + e^{-\lambda \cdot d(w,t)^2}}{2}, \tag{3}$$

where λ is a parameter to control the decrease degree of the function value with the increase of distance $d(w,t)$. If λ is large, then the quality decreases quickly with the increase of distance. For example, Figure 4 shows three functions with λ of 100, 10 and 0.1 respectively. If $\lambda = 100$, then the quality becomes 0.5 when the distance is 0.2. On the contrary, if $\lambda = 0.1$, the quality is still above 0.9 when the distance is 1.0.

The reason that we use the bell-shaped function is threefold: (i) we aim to model the quality as the probability that the worker gives correct answers, and the function value is within [0,1] which is coincident with probability values. Moreover, we set the minimum value of the quality as 0.5, because the worst probability for a worker is to randomly give an answer, which is 0.5. (ii) The function decreases exponentially with the increase of distance. (iii) The decrease rate can be well controlled by the parameter λ.

However, simply using a single distance function with an unknown parameter λ has two limitations. First, the expressiveness of modeling the quality with a single function is weak, because the quality may coincidentally regress around a single function. Second, it is difficult to learn the non-random variable parameter λ in a probability model, because there is no closed-form solution to compute λ directly. To address these problems, we propose the *distance function set* and use it to model the distance-aware quality.

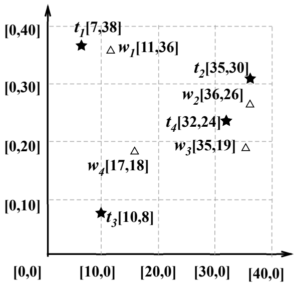
Definition 4 (Distance-Function Set): The distance function set $\mathcal{F}$ consists of a set of bell-shaped functions with fixed parameters $\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{F}|}$, i.e.,

$$\mathcal{F} = \{f_{\lambda_1}, f_{\lambda_2}, \ldots, f_{\lambda_{|\mathcal{F}|}}\}. \tag{4}$$

For example, Figure 4 shows a distance function set with $\mathcal{F} = \{f_{100}, f_{10}, f_{0.1}\}$.

Definition 5 (Distance-aware Quality): The distance-aware quality of a worker w on a task t is a combination of distance functions, i.e.,

[2] $d(w,t)$ is normalized by a maximum distance (e.g. the maximum distance between POIs). Note that a worker may submit multiple locations for POI labelling tasks, e.g., home, office, intertested zones. To this end, we measure the distance by using the minimum distance from his locations to tasks as we assume the worker may be familiar with nearby POIs around all his locations.



W	$\mathcal{R}(w,t)$	$P(i_w=1)$	$P(d_w)$	T	$P(z_{t,k}=1)$	P(d_t)
w_1	t_1:[1,1,0], t_4:[1,0,0]	0.89	[0.07,0.12,0.81]	t_1	[0.64,0.64,0.35]	[0.10,0.21,0.69]
w_2	t_2:[1,1,0], t_3:[1,1,0]	0.93	[0.04,0.09,0.87]	t_2	[0.72,0.72,0.25]	[0.04,0.06,0.90]
w_3	t_2:[1,1,0], t_3:[1,0,0]	0.93	[0.05,0.06,0.89]	t_3	[0.71,0.49,0.28]	[0.06,0.07,0.87]
w_4	t_2:[0,0,0], t_4:[0,1,1]	0.19	[0.41,0.40,0.19]	t_4	[0.59,0.40,0.40]	[0.24,0.24,0.52]

Fig. 3. A Running Example

$$\mathtt{DQ}_w = \sum_{f_{\lambda_i}\in\mathcal{F}} P(d_w{=}f_{\lambda_i}) \cdot f_{\lambda_i}(d(w,t)), \qquad (5)$$

where d_w is a random variable which satisfies a multinomial distribution over the set and $P(d_w = f_\lambda)$ can be treated as the weight of f_λ in the function set. Since different workers have different distributions, d_w clearly reflects the different influence of distance towards workers' qualities. For example, for the distance function set in Figure 4, if $P(d_w = f_{100}) = 0.6$, $P(d_w = f_{10}) = 0.2$, $P(d_w = f_{0.1}) = 0.2$, then w can possibly provide accurate answers only for nearby POIs. Otherwise, if $P(d_w = f_{100}) = 0.2$, $P(d_w = f_{10}) = 0.2$, $P(d_w = f_{0.1}) = 0.6$, then w is able to provide accurate answers for distant POIs. In Figure 4, if $P(d_w = f_{100}) = P(d_w = f_{10}) = P(d_w = f_{0.1}) = \frac{1}{3}$, then the distance-aware quality is shown as the dash line in the figure.

POI-Influence. Similarly, we model the POI-influence based on the distance function set.

Definition 6 (POI-Influence Quality): We define the POI-influence quality $\mathtt{IQ}_t$ as a combination of distance functions

$$\mathtt{IQ}_t = \sum_{f_{\lambda_i}\in\mathcal{F}} P(d_t = f_{\lambda_i}) \cdot f_{\lambda_i}(d(w,t)). \qquad (6)$$

where d_t is also a random variable with multinomial distribution over the set and $P(d_t = f_{\lambda_i})$ is the weight of f_{λ_i} in the function set for d_t. For the distance function set in Figure 4, if a POI has a large influence, then $P(d_t = f_{0.1})$ is large and $P(d_t = f_{100})$ is small.

Answer Accuracy. Given a task t and a worker w, suppose w returns $r_{w,t,k}$ for label $l_{t,k}$, we model the accuracy of the answer $r_{w,t,k}$ as the probability of $r_{w,t,k}$ being a true result $z_{t,k}$. We consider two cases to compute the probability.

Case 1: if w is an unqualified worker, i.e., $i_w = 0$, then w randomly gives a 1/0 answer with a probability of 0.5 to be the true result. Therefore, we have

$$P(r_{w,t,k} = z_{t,k} \mid i_w = 0) = 0.5. \qquad (7)$$

Case 2: if w is a well qualified worker, the probability of $r_{w,t,k}$ being a correct label is determined by both the distance-aware quality of w and POI-influence, which can be computed as their linear combination with the following equation:

$$\begin{aligned} P(r_{w,t,k}{=}z_{t,k}|i_w{=}1) &= \alpha \cdot \sum_{f_{\lambda_i}\in\mathcal{F}} P(d_w{=}f_{\lambda_i}) \cdot f_{\lambda_i}(d(w,t)) \\ &+ (1-\alpha) \cdot \sum_{f_{\lambda_i}\in\mathcal{F}} P(d_t{=}f_{\lambda_i}) \cdot f_{\lambda_i}(d(w,t)), \end{aligned} \qquad (8)$$

where we use a constant α (e.g. 0.5) to tune the weight of distance-aware quality and POI-influence.

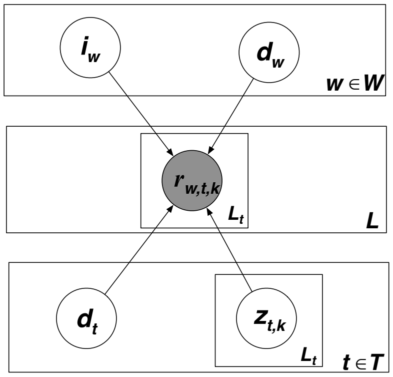


Fig. 5. Inference Model

Combining these two cases, we generate the probability of $r_{w,t,k}$ being a true result $z_{t,k}$:

$$\begin{aligned} P(r_{w,t,k}{=}z_{t,k}) &= P(r_{w,t,k}{=}z_{t,k}|i_w{=}0) \cdot P(i_w{=}0) \\ &+ P(r_{w,t,k}{=}z_{t,k}|i_w{=}1) \cdot P(i_w{=}1). \end{aligned} \qquad (9)$$

In summary, the accuracy of an answer $r_{w,t,k}$ is determined by w's inherent quality and distance-aware quality, as well as the POI-influence. Next we use a graphical model to formally define our model.

Graphical Probability Model. We show the graphical description of our model in Figure 5. Each node (i.e., $z_{t,k}$, i_w, d_w, d_t, and $r_{w,t,k}$) in the graph represents a random variable and the shaded node $r_{w,t,k}$ indicates the corresponding answers given by workers. The arrows from $z_{t,k}$, i_w, d_w, d_t to $r_{w,t,k}$ indicate that $r_{w,t,k}$ is generated based on a distribution conditioned on $z_{t,k}$, i_w, d_w, d_t. The generative process of $r_{w,t,k}$ is as follows:

- For each task t:
  - ○ For each label $l_{t,k}$: Generate $z_{t,k}$ with a Bernoulli distribution
  - ○ Generate d_t with a multinomial distribution
- For each worker t:
  - ○ Generate i_w with a Bernoulli distribution
  - ○ Generate d_w with a multinomial distribution
- For each answer $r_{w,t,k}$:
  - ○ Generate $r_{w,t,k}$ with the distribution $P(r_{w,t,k}|z_{t,k})$

Notice that the distribution $P(r_{w,t,k}|z_{t,k})$ is related to the value of $z_{t,k}$, i_w, d_w and d_t. In fact, it is a simple deduction of the modeled answer accuracy introduced in Equation 9:

$$\begin{aligned} P(r_{w,t,k}{=}1|z_{t,k}) &= P(r_{w,t,k}{=}z_{t,k}{=}1) \cdot P(z_{t,k}{=}1) \\ &+ P(r_{w,t,k}{\neq}z_{t,k}{=}0) \cdot P(z_{t,k}{=}0), \\ P(r_{w,t,k}{=}0|z_{t,k}) &= 1 - P(r_{w,t,k}{=}1|z_{t,k}). \end{aligned} \qquad (10)$$

To this end, we can estimate the true results of labels $(P(z_{t,k}))$, the worker quality $(P(i_w), P(d_w))$ and the POI-influence $(P(d_t))$ by learning the model. Moreover, given a worker w and a task t, we can estimate the accuracy of the answer as $P(z_{t,k} = r_{w,t,k})$ based on Equation 9. We call $P(z_{t,k}), P(i_w), P(d_w), P(d_t)$ as the parameters of our model and next we introduce how to estimate them with an Expectation Maximization (EM) [6] method in Section III-C.

Example 1: Consider the example in Figure 3. We have four workers and four tasks, whose locations are shown in the left figure. Each worker has done two tasks in the right table. We show the estimated parameters in the table, which can be derived by running the EM method (see Section III-C). We find that w_2 and w_3 have the best inherent quality and strong distance-based quality (i.e. the distance can hardly affect their quality). On the other hand, w_4 has a low-inherent quality and weak distance-based quality. This result is coincident with the workers' answers to tasks as w_4 always gives conflicting answers to the other workers for tasks t_2 and t_3. The inference results of t_1, t_2, t_3, t_4 are in column $P(z_{t,k} = 1)$ and their POI-influences are in column $P(d_t)$. Based on the estimated parameters and Equation 9, we can estimate the probability of w_2 giving accurate answers to t_4 as $P(z_{t_4,k} = r_{w_2,t_4,k}) = 0.87$.

C. Parameter Estimation

We use the maximum likelihood estimation (MLE) to estimate the parameters. As all the workers' answers ($r_{w,t,k}$) are independent, it aims to maximize

$$\textsf{argmax}_{z_{t,k},i_w,d_w,d_t} \prod_{t,w,l} P(r_{w,t,k})$$

$$= \prod_{t,w,l} \sum_{z_{t,k},i_w,d_w,d_t} P(r_{w,t,k}|z_{t,k}, i_w, d_w, d_t)P(i_w)P(z_{t,k})P(d_w)P(d_t) \tag{11}$$

As the likelihood function follows a sum-product form, the parameters cannot be directly obtained from its derivation. Thus we utilize an EM method [6], which iteratively estimates the parameters through E-step and M-step.

E-step. It assumes all the values of parameters as known and computes the conditional probability of unobserved variables over observed ones. In our model, we need to consider four cases as $i_w = 0/i_w = 1$ and $z_{t,k} = 1/z_{t,k} = 0$ follow different distributions. The conditional probability is computed as

$P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$
$\propto P(r_{w,t,k}|z_{t,k}, i_w, d_w, d_t)P(i_w)P(z_{t,k})P(d_w)P(d_t)$

Case 1: For $i_w = 0$ and $z_{t,k} = 0$, we have
$P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$
$\propto P(r_{w,t,k}{=}z_{t,k}|i_w{=}0)P(z_{t,k})P(i_w)P(d_w)P(d_t)$

Case 2: For $i_w = 0$ and $z_{t,k} = 1$, we have
$P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$
$\propto P(r_{w,t,k} \neq z_{t,k}|i_w{=}0)P(z_{t,k})P(i_w)P(d_w)P(d_t)$

Case 3: For $i_w = 1$ and $z_{t,k} = 0$, we have
let $q(d_w, d_t) = \alpha f_{d_w}(d(w,t)) + (1-\alpha) f_{d_t}(d(w,t))$,
$P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$
$= q(d_w, d_t)^{1\text{-}r_{w,t,k}}(1\text{-}q(d_w, d_t))^{r_{w,t,k}} P(z_{t,k})P(i_w)P(d_w)P(d_t)$

Case 4: For $i_w = 1$ and $z_{t,k} = 1$, we have
$P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$
$\propto q(d_w, d_t)^{r_{w,t,k}}(1\text{-}q(d_w, d_t))^{1\text{-}r_{w,t,k}} P(z_{t,k})P(i_w)P(d_w)P(d_t)$ (12)

M-step. It estimates the parameters by maximizing the expectation of the log-likelihood of all the variables, i.e.

$$\textsf{maximize} \sum_{t,w,l} \mathbb{E}_{[P(z_{t,k},i_w,d_w,d_t|r_{w,t,k})]} \ln P(r_{w,t,k}, z_{t,k}, i_w, d_w, d_t) \tag{13}$$

where E is the expected log-likelihood.

By setting the derivation of the log-likelihood on all parameters as zero, $P(z_{t,k})$, $P(i_w)$, $P(d_w)$, $P(d_t)$ can then be deduced as the respective marginal distribution over the conditional distribution $P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$. Thus we can simply estimate them by summing up other parameters in $P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})$. Formally, we have

$$\begin{aligned}
P(z_{t,k}) &= \frac{\sum\limits_{w \in W(t)} \sum\limits_{i_w,d_w,d_t} P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})}{|W(t)|} \\
P(i_w) &= \frac{\sum\limits_{t \in T(w)} \sum\limits_{l_{t,k} \in L_t} \sum\limits_{z_{t,k},d_w,d_t} P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})}{\sum_{t \in T(w)} |L_t|} \\
P(d_w) &= \frac{\sum\limits_{t \in T(w)} \sum\limits_{l_{t,k} \in L_t} \sum\limits_{z_{t,k},i_w,d_t} P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})}{\sum_{t \in T(w)} |L_t|} \\
P(d_t) &= \frac{\sum\limits_{t \in T(w)} \sum\limits_{l_{t,k} \in L_t} \sum\limits_{z_{t,k},i_w,d_w} P(z_{t,k}, i_w, d_w, d_t|r_{w,t,k})}{\sum_{t \in T(w)} |L_t|}
\end{aligned} \tag{14}$$

Time Complexity. For each iteration, the algorithm computes the conditional probability for all answers in E-step and reuses them in M-step with cost $O(B \cdot |L_t|)$, where B is the total number of answers. Suppose the total number of iterations is $\mathcal{I}$, then the time complexity of the EM method is $O(B \cdot |L_t| \cdot \mathcal{I})$.

D. Model Update

When a worker returns an answer, we need to update the parameters. It could be expensive to run the EM algorithm for every answer submission. Therefore, we update the model in two ways. First, we can use the complete EM algorithm in a delayed manner, e.g., we run the complete EM algorithm only if there are 100 submissions. Second, during each interval, we utilize the incremental EM algorithm [18] to update the parameters. The incremental EM algorithm only updates the quality of the workers who have done the task, and updates both the inferred results and the POI-influence for those tasks that have been assigned to the worker (Equations 12 and 14).

IV. Task Assignment

In this section, we study how to assign appropriate tasks to available workers in W that ask for tasks. For each task, we consider how much accuracy will be improved if it is assigned to some workers in W. Then we select the best tasks for each worker to maximize the accuracy improvement.

A. Assignment Overview

In our model, $P(z_{t,k})$ is used to infer the result of $l_{t,k}$. If the true result of $l_{t,k}$ is 1 (i.e., $z_{t,k} \equiv 1$), the accuracy of our inference is $P(z_{t,k} = 1)$; if the true result is 0 (i.e., $z_{t,k} \equiv 0$), the accuracy of our inference is $P(z_{t,k} = 0)$.

Formally we denote $Acc_{t,k}$ as the accuracy of our inference on a label $l_{t,k}$, where

$$Acc_{t,k} = \begin{cases} P(z_{t,k} = 1) & z_{t,k} \equiv 1 \\ P(z_{t,k} = 0) & z_{t,k} \equiv 0. \end{cases} \tag{15}$$

In task assignment, if the task t is assigned to a set of new workers, denoted by $\widehat{W}(t) \subseteq W$, $Acc_{t,k}$ will change as the answer set of t changes. Let $Acc_{t,k}(\widehat{W}(t))$ denote the accuracy of t after workers in $\widehat{W}(t)$ submit their answers. Intuitively, we will assign t to workers in $\widehat{W}(t)$ if the accuracy improvement (i.e., $\Delta Acc_{t,k}(\widehat{W}(t)) = Acc_{t,k}(\widehat{W}(t)) - Acc_{t,k}$) is large. Thus we need to know how much quantity $Acc_{t,k}(\widehat{W}(t))$ will change if t is assigned to $\widehat{W}(t)$. To address this problem, we first discuss how to estimate the accuracy $Acc_{t,k}(\widehat{W}(t))$ in Section IV-B. As the accuracy $Acc_{t,k}(\widehat{W}(t))$ is dependent on the true result of $l_{t,k}$ (which is not known to us), we use its expected accuracy improvement instead, and then propose an assignment algorithm to maximize the expected accuracy improvement in Section IV-C.

B. Accuracy Estimation

We discuss how to predict the accuracy of a task if the task is assigned to some workers.

Estimation for a Single Worker. We first consider the case that t is assigned to a single worker w, i.e., $\widehat{W}(t) = \{w\}$. Let $Acc_{t,k}(w)$ denote the inferred accuracy after the answer $r_{w,t,k}$ from worker w on task $l_{t,k}$ is submitted. Next we introduce how to estimate $Acc_{t,k}(w)$.

Since $Acc_{t,k}(w)$ is based on the result of $l_{t,k}$, we need to consider the case $z_{t,k} \equiv 1$ and $z_{t,k} \equiv 0$ separately.

(1) For the case $z_{t,k} \equiv 1$, we can derive $Acc_{t,k}(w) = P(z_{t,k} = 1|r_{w,t,k})$, where $P(z_{t,k} = 1|r_{w,t,k})$ is the inference result after $l_{t,k}$ is answered by w with the answer $r_{w,t,k}$. Based on our inference model in Equation 14, we have

$$P(z_{t,k}{=}1|r_{w,t,k}){=}\frac{\sum\limits_{w \in W(t) \cup \{w\}} \sum\limits_{i_w, d_w, d_t} P(z_{t,k}{=}1, i_w, d_w, d_t | r_{w,t,k})}{|W(t)| + 1}$$
$$= \frac{|W(t)| \cdot P(z_{t,k}{=}1) + \sum\limits_{i_w, d_w, d_t} P(z_{t,k}{=}1, i_w, d_w, d_t | r_{w,t,k})}{|W(t)| + 1}$$
$$= \frac{|W(t)| \cdot P(z_{t,k}{=}1) + P(z_{t,k}{=}1 | r_{w,t,k})}{|W(t)| + 1}.$$

To compute $P(z_{t,k} = 1|r_{w,t,k})$, we need to know the exact value of $r_{w,t,k}$. If $r_{w,t,k} = 1$, we have $r_{w,t,k} = z_{t,k}$; otherwise, $z_{t,k} \neq r_{w,t,k}$. To conclude we have

$$P(z_{t,k} = 1|r_{w,t,k}) = \begin{cases} P(z_{t,k}{=}r_{w,t,k}) & r_{w,t,k} = 1 \\ P(z_{t,k}{\neq}r_{w,t,k}) & r_{w,t,k} = 0, \end{cases}$$

where $P(z_{t,k}{=}r_{w,t,k})$ (or $P(z_{t,k} \neq r_{w,t,k})$) is the answer accuracy estimated based on our model in Equation 9 [3].

As $r_{w,t,k}$ cannot be obtained before assignment, we can only compute the expected value of $P(z_{t,k} = 1|r_{w,t,k})$. If $P(z_{t,k} \equiv 1)$, then $P(r_{w,t,k} = 1) = P(r_{w,t,k} = z_{t,k})$ and $P(r_{w,t,k} = 0) = P(r_{w,t,k} \neq z_{t,k})$, thus we have

$$\begin{aligned} &P_{\mathbb{E}}(z_{t,k}{=}1|r_{w,t,k}){=}P(z_{t,k}{=}1|r_{w,t,k}{=}1)P(r_{w,t,k}{=}1) \\ &+ P(z_{t,k}{=}1|r_{w,t,k}{=}0)P(r_{w,t,k}{=}0) \\ &{=}\frac{|W(t)| \cdot P(z_{t,k}{=}1){+}P(z_{t,k}{=}r_{w,t,k})}{|W(t)|{+}1} \cdot P(z_{t,k}{=}r_{w,t,k}) \\ &{+}\frac{|W(t)| \cdot P(z_{t,k}{=}1){+}P(z_{t,k}{\neq}r_{w,t,k})}{|W(t)|{+}1} \cdot P(z_{t,k}{\neq}r_{w,t,k}). \end{aligned} \tag{16}$$

(2) For the case $z_{t,k} \equiv 0$, similarly $Acc_{t,k}(w) = P(z_{t,k} = 0|r_{w,t,k})$ and the expected value of $P(z_{t,k} = 0|r_{w,t,k})$ is

$$\begin{aligned} &P_{\mathbb{E}}(z_{t,k} = 0|r_{w,t,k}) = \\ &\frac{|W(t)| \cdot P(z_{t,k}{=}0){+}P(z_{t,k}{=}r_{w,t,k})}{|W(t)|{+}1} \cdot P(z_{t,k}{=}r_{w,t,k}) \\ &{+}\frac{|W(t)| \cdot P(z_{t,k}{=}0){+}P(z_{t,k}{\neq}r_{w,t,k})}{|W(t)|{+}1} \cdot P(z_{t,k}{\neq}r_{w,t,k}). \end{aligned} \tag{17}$$

To conclude, we compute $Acc_{t,k}(w)$ as the expected probability of $P(z_{t,k} = 1|r_{w,t,k})$ and $P(z_{t,k} = 0|r_{w,t,k})$ with the following equation:

$$\begin{aligned} &Acc_{t,k}(w) = \begin{cases} P_{\mathbb{E}}(z_{t,k} = 1|r_{w,t,k}) & (z_{t,k} \equiv 1) \\ P_{\mathbb{E}}(z_{t,k} = 0|r_{w,t,k}) & (z_{t,k} \equiv 0) \end{cases} \\ &= \frac{|W(t)| \cdot Acc_{t,k} + P(z_{t,k}{=}r_{w,t,k})}{|W(t)| + 1} \cdot P(z_{t,k}{=}r_{w,t,k}) \\ &+ \frac{|W(t)| \cdot Acc_{t,k} + P(z_{t,k}{\neq}r_{w,t,k})}{|W(t)| + 1} \cdot P(z_{t,k}{\neq}r_{w,t,k}). \end{aligned} \tag{18}$$

Example 2: Consider the first label $l_{t_4,1}$ of task t_4 in Figure 3. Based on our current inference we have $P(z_{t_4,1} = 1) = 0.59$, $P(z_{t_4,1} = 0) = 0.41$. Suppose t_4 is assigned to w_2, the estimated accuracy of w_2 to t_4 is $P(z_{t_4,1} = r_{w_2,t_4,1}) = 0.87$. To calculate the estimated accuracy, if $z_{t,k} \equiv 1$, then $P_{\mathbb{E}}(z_{t_4,1} = 1|r_{w_2,t,1}) = \frac{2\times0.59+0.87}{2+1} \times 0.87 + \frac{2\times0.59+0.13}{2+1} \times 0.13 = 0.65$. Similarly if $z_{t,k} \equiv 0$, $P_{\mathbb{E}}(z_{t_4,1} = 0|r_{w_2,t,1}) = \frac{2\times0.41+0.87}{2+1} \times 0.87 + \frac{2\times0.41+0.13}{2+1} \times 0.13 = 0.53$.

Estimation for Multiple Workers. We discuss how accuracy changes when more than one worker give answers to $l_{t,k}$. We first prove that the sequence of workers' answers to $l_{t,k}$ do not affect the estimated accuracy with the following lemma:

Lemma 1: Let $Acc_{t,k}(w_1, w_2)$ denote the accuracy of $l_{t,k}$ after w_1 and w_2 give answers to $l_{t,k}$. We have $Acc_{t,k}(w_1, w_2) = Acc_{t,k}(w_2, w_1)$

Proof: See appendix. ■

Based on Lemma 1, when more than one worker give answers to $l_{t,k}$, its accuracy stays the same regardless of the sequence of answers. Suppose t is assigned to a set of workers $\widehat{W}(t)$, we do not need to consider the sequence of these workers. Next we introduce how to estimate $Acc_{t,k}(\widehat{W}(t))$.

Recall that $Acc_{t,k}(\widehat{W}(t))$ is an expected probability based on the value of $r_{w,t,k}$ ($w \in \widehat{W}$). To compute $Acc_{t,k}(\widehat{W}(t))$, we have to enumerate all possible combinations of $r_{w,t,k}$

[3] If w is a new worker or t has not been answered by any worker, we cannot compute $P(z_{t,k}{=}r_{w,t,k})$ based on current inference. To handle those workers or tasks, we simply assume they have the best worker quality (e.g., $P(i_w = 1) = 1$, $P(d_w = f_{0.1}) = 1$) and the largest POI-influence (e.g., $P(d_t = f_{0.1}) = 1$) as we aim to estimate their "real" quality as soon as possible by giving priorities for those workers and tasks on task assignment.

with $O(2^{|\widehat{W}(t)|})$ time. Fortunately we find that $Acc_{t,k}(\widehat{W}(t))$ satisfies the following recursive property (Lemma 2) and thereby $Acc_{t,k}(\widehat{W}(t))$ can be calculated in linear time.

Lemma 2: $Acc_{t,k}(\widehat{W}(t))$ can be recursively computed by $Acc_{t,k}(\widehat{W}(t) - \{w\})$ as

$$Acc_{t,k}(\widehat{W}(t))= \begin{cases} P_{\mathbb{E}}(z_{t,k} = 1|\widehat{W}(t)) & (z_{t,k} \equiv 1) \\ P_{\mathbb{E}}(z_{t,k} = 0|\widehat{W}(t)) & (z_{t,k} \equiv 0) \end{cases}$$
$$= \frac{(|W(t)|+|\widehat{W}(t)|\text{-}1) \cdot Acc_{t,k}(\widehat{W}(t)\text{-}\{w\})P(z_{t,k}{=}r_{w,t,k})}{|W(t)|+|\widehat{W}(t)|} P(z_{t,k}{=}r_{w,t,k})$$
$$+ \frac{(|W(t)|+|\widehat{W}(t)|\text{-}1) \cdot Acc_{t,k}(\widehat{W}(t)\text{-}\{w\})+P(z_{t,k}{\neq}r_{w,t,k})}{|W(t)|+|\widehat{W}(t)|} P(z_{t,k}{\neq}r_{w,t,k}), \tag{19}$$

where w is an arbitrary worker in $\widehat{W}(t)$.

Proof: See appendix. ■

Based on Lemma 2, we can estimate $Acc_{t,k}(\widehat{W}(t))$ in linear time $O(\widehat{W}(t))$. Suppose $\widehat{W}(t)$=$\{w_1, w_2, \ldots, w_{|\widehat{W}(t)|}\}$, we can first compute $Acc_{t,k}(w_1)$, then compute $Acc_{t,k}(w_1, w_2)$ based on $Acc_{t,k}(w_1)$ with $O(1)$ time and repeat the computation until we get $Acc_{t,k}(w_1, w_2, \ldots, w_{|\widehat{W}(t)|})$.

Example 3: Consider the first label $l_{t_4,1}$ of task t_4 in Figure 3. If both workers w_2 and w_3 give answers to t_4, we have $P_{\mathbb{E}}(z_{t_4,1} = 1|r_{w_2,t,1}) = 0.65$ and $P_{\mathbb{E}}(z_{t_4,1} = 0|r_{w_2,t,1}) = 0.53$. Based on our inference model, the estimated accuracy of w_3 to t_4 is $P(z_{t_4,1} = r_{w_3,t_4,1}) = 0.86$. Thus if $z_{t,k} \equiv 1$, $P_{\mathbb{E}}(z_{t_4,1} = 1|r_{w_2,t,1}, r_{w_3,t,1}) = \frac{0.65\times3+0.86}{4} \times 0.86 + \frac{0.65\times3+0.14}{4} \times 0.14 = 0.69$. Similarly, $P_{\mathbb{E}}(z_{t_4,1} = 0|r_{w_2,t,1}, r_{w_3,t,1})$=$\frac{0.53\times3+0.86}{4} \times 0.86$+$\frac{0.53\times3+0.14}{4} \times 0.14$=$0.61$.

C. Optimal Task Assignment

Optimal Task Assignment Problem. Based on the estimated accuracy, next we introduce the optimal task assignment problem. For the available worker set W, our goal is to find an assignment $\mathcal{A}(W)$ to maximize the overall accuracy improvement. However, as the accuracy $Acc_{t,k}$ depends on the true result of $l_{t,k}$ for which we do not know, we compute an expected accuracy improvement based on the current probability of $z_{t,k}$. The expected accuracy improvement is

$$\Delta Acc_{t,k}(\widehat{W}(t)){=}P(z_{t,k}{=}1) \cdot (P_{\mathbb{E}}(z_{t,k}{=}1|\widehat{W}(t)) - P(z_{t,k}{=}1)) + P(z_{t,k}{=}0) \cdot (P_{\mathbb{E}}(z_{t,k}{=}0|\widehat{W}(t)) - P(z_{t,k}{=}0)). \tag{20}$$

Formally, we define the optimal task assignment problem.

Definition 7 (Optimal Task Assignment): The optimal task assignment is to find an assignment that maximizes the overall expected accuracy improvement, i.e.,

$$\text{argmax}_{\mathcal{A}(w), w\in W} \sum_{t\in T} \sum_{k=1}^{|L_t|} \Delta Acc_{t,k}(\widehat{W}(t)) \quad \text{s.t. } |\mathcal{A}(w)| = h \tag{21}$$

where $\mathcal{A}(w)$ is the set of h tasks assigned to worker w, and $\widehat{W}(t) = \{w|t \in \mathcal{A}(w)\}$ is the set of workers assigned with t.

Algorithm 1: GreedyAssignment

```
   Input: $W$: a set of available workers for tasks.
   Output: $\mathcal{A}(W)$: assigned tasks for workers in $W$.
1  $\mathcal{A}(W) = \{\mathcal{A}(w) = \emptyset \mid w \in W\}$;
2  foreach $t \in T$ do
3      $\widehat{W}(t) = \emptyset$;
4  foreach $w \in W$ do
5      foreach $t \in T$ do
6          for $k = 1$ to $|L_t|$ do
7              $Acc[w][t][k] = Acc_{t,k}(w)$;
8          $\Delta Acc[w][t] = \sum_{k=1}^{|L_t|} \Delta Acc_{t,k}(w)$;
9  while $|\mathcal{A}(W)| < h \cdot |W|$ do
10     $t_{max}, w_{max} = \arg\max_{t,w} \Delta Acc$;
11     $\mathcal{A}(w_{max}).append(t_{max})$;
12     $\widehat{W}(t_{max}).append(w_{max})$;
13     $\Delta Acc[w_{max}].remove(t_{max})$;
14     if $|\mathcal{A}(w_{max})| \geq h$ then
15         $\Delta Acc.remove(w_{max})$;
16     foreach $w \in W - \widehat{W}(t_{max})$ do
17         for $k = 1$ to $|L_t|$ do
18             $Acc[w][t][k] = Acc_{t,k}(\widehat{W}(t) \cup \{w\})$;
19         $\Delta Acc[w][t_{max}] = \sum_{k=1}^{|L_t|} \Delta Acc_{t,k}(\widehat{W}(t_{max}) \cup \{w\})$;
20 return $\mathcal{A}(W)$;
```

Unfortunately, we find the problem of finding the optimal assignment to maximize the increase of accuracy is NP-hard (Lemma 3). Then we propose a greedy algorithm.

Lemma 3: The Optimal Assignment Problem is NP-hard.

Proof: See Appendix. ■

A Greedy Algorithm. The algorithm greedily picks a pair (task, worker) with maximum increase of accuracy until each worker in W has been assigned h tasks. Algorithm 1 shows the details of the algorithm. It first initializes $\mathcal{A}(W)$ and $\widehat{W}(t)$ as empty set (lines 1-3). Then it computes all the estimated accuracy and keeps them in a matrix *Acc*, where $Acc[w][t][k]$ is $Acc_{t,k}(w)$ computed in Equation 18 (line 7). Notice that in practice, $Acc[w][t][k]$ stores a pair of values ($P_{\mathbb{E}}(z_{t,k} = 1|r_{w,t,k})$, $P_{\mathbb{E}}(z_{t,k} = 0|r_{w,t,k})$). Meanwhile, it also initializes a matrix ΔAcc to record all the accuracy improvement, where $\Delta Acc[w][t]$ is the accuracy improvement $\Delta Acc_{t,k}(\widehat{W}(t))$ computed in Equation 20 if t is assigned to w (line 8). At each iteration, we pick the pair of worker and task (w_{max}, t_{max}) with the maximum accuracy improvement from matrix ΔAcc and put them into $\mathcal{A}(W)$ and $\widehat{W}(t)$ (lines 10-13). If h tasks have been assigned to worker w_{max}, it removes w_{max} from the matrix ΔAcc (line 15) to avoid duplicate assignments. As workers in $\widehat{W}(t_{max})$ have been assigned with t_{max}, for the rest workers in $W - \widehat{W}(t_{max})$, we update matrix

Acc and ΔAcc for t_{max} by assuming that w is further assigned with t_{max} (line 18-19). The algorithm terminates when all workers have been assigned with h tasks.

Time Complexity. To initialize the matrices of Acc and ΔAcc, we need to compute the estimated accuracy for each (worker, task) pair, the cost is $O(|W| \cdot |T| \cdot |L_t|)$. At each iteration, we update Acc and ΔAcc of t_{max} for every worker, and the cost is $O(|W| \cdot |L_t|)$. As the total number of iterations is $h \cdot |W|$, the time complexity is $O(|W| \cdot |T| \cdot |L_t| + h \cdot |W|^2 \cdot |L_t|)$.

Example 4: Consider the example in Figure 3. Suppose $W = \{w_2\}$ and $h = 1$, for label $l_{t_4,1}$, we have computed $P_{\mathbb{E}}(z_{t_4,1} = 1 | r_{w_2,t,1}) = 0.65$ and $P_{\mathbb{E}}(z_{t_4,1} = 0 | r_{w_2,t,1}) = 0.53$ in Example 2. Therefore, its expected accuracy improvement is $\Delta Acc_{t_4,1}(w_2) = 0.59 \times (0.65 - 0.59) + 0.41 \times (0.53 - 0.41) = 0.08$. Similarly, we can compute $\Delta Acc_{t_4,2}(w_2) = \Delta Acc_{t_4,3}(w_2) = 0.08$. The greedy algorithm will first assign t_4 to w_2 as it provides maximum accuracy improvement. It can be seen that it is beneficial to assign t_4 to w_2. The reason is that previous workers w_1 and w_4 have returned completely different answers on t_4, while w_2 can provide high-quality answers to improve inference. Our algorithm can judiciously capture this through maximizing the accuracy improvement.

V. Experimental Study

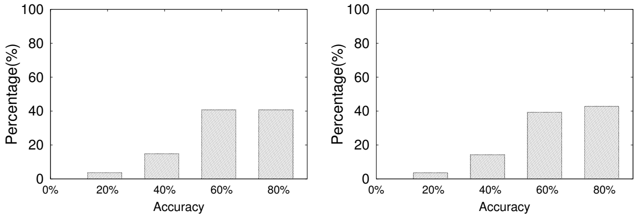
A. Experiment Setup

Datasets. We used two real datasets called Beijing and China as our task sets. The Beijing dataset contained 200 POIs with their locations in Beijing, including parks, universities, restaurants, etc. The China dataset contained 200 scenic spots in China, e.g. "Tiananmen Square", "Oriental Pearl Tower", etc. For each task, we set the number of labels $|L_t| = 10$. To generate correct labels as the ground truth, we collected labels from Dianping[4]. For each task, we randomly selected 1∼10 correct labels and manually checked their correctness and then complemented the label set with incorrect labels. Beijing and China contained 927/1073 and 864/1136 correct/incorrect labels respectively.

Experiment Deployment. We conducted our experiment on ChinaCrowds[5], the largest Chinese crowdsourcing platform. It had mobile applications which supported location-based tasks by locating workers with GPS equipment. In our experiment, workers were asked to select and submit one or several familiar locations with geo-coordinate to do the POI labelling tasks. We deployed two parts of experiments and used 1000 budget (0.2 RMB for each task) for each dataset. For each assignment, we assign h=2 tasks to each worker.

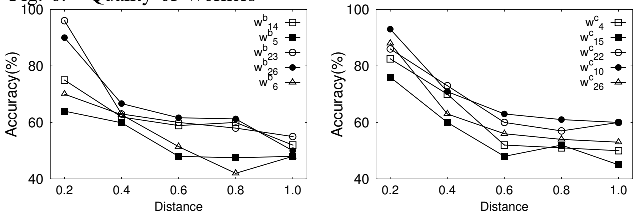
Deployment 1 - Evaluation of inference models. The tasks were published on the platform and each task was answered by five workers. Then we analyzed the quality of workers (influence of POIs) and compared our inference model with baselines (as described later) based on the collected 2000 assignments.

Deployment 2 - Evaluation of task assignments. We adopted the developer mode in ChinaCrowds, enabling us to assign specific tasks to those workers based on our own developed assignment algorithms when online workers requested tasks.



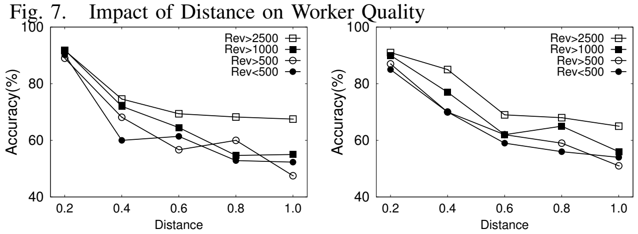
(a) Beijing (b) China

Fig. 6. Quality of Workers



(a) Beijing (b) China

Fig. 7. Impact of Distance on Worker Quality



(a) Beijing (b) China

Fig. 8. Impact of Distance on the POI-Influence

Baselines. For evaluation of inference models, we compared our Inference Model (IM) with two widely used baselines: (1) the Majority Voting (MV) method and (2) the Expectation-Maximization (EM) algorithm [5]. MV determined the labels based on the majority answers from workers. EM iteratively estimated each worker's quality (confusion matrix) and exploited the estimated quality to infer the correct labels. To evaluate the task assignment algorithms, we compared our AccOpt with (1) Random and (2) a Spatial-First assignment algorithm (SF). The Random algorithm randomly assigned tasks to online workers. The SF algorithm optimized the distance between workers and tasks. For each online worker w, it assigned the closest undone task(s) to w.

We implemented all the methods in Python and ran experiments on a Ubuntu machine with 16GB RAM, Intel Xeon CPU 2.93GHz. We set α=0.5, $\mathcal{F}$ = [$f_{0.1}$, f_{10}, f_{100}] for IM.

Evaluation Metrics. We evaluated the effectiveness and efficiency of our proposed methods in term of the accuracy (described in Sec. II) and running time respectively.

B. Data Analysis

We first analyzed the collected answers to verify our intuitions. For each answer, we computed its accuracy as the percentage of correctly answered labels over the ground truth, and reported the average accuracy across all answers.

First, we tested the quality of workers from a general perspective. To eliminate the impact of distance, we collected the answers that workers and tasks were within a distance of 0.2 and computed their average accuracy on those answers. We reported the percentages (with five ranges from [1%,20%] to $[81\%, 100\%]$) of their average accuracy in Figure 6. We could see that although workers and tasks were very close,

[4] http://www.dianping.com/

[5] http://www.chinacrowds.com/

the quality of workers differed. Most workers returned high-quality answers for spatially nearby tasks. For example in the China dataset, most workers returned answers with accuracy over 60%; however, there were about 20% workers returning low-quality answers with accuracy under 60%. In our model, this was attributed to the inherent quality of workers where the accuracy of answers from those with low inherent quality was low even when the distance was short.

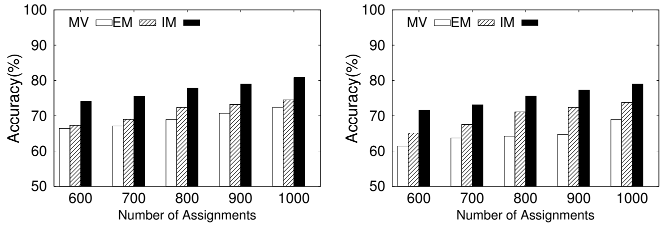
Next, we tested the impact of the distance on workers' qualities. We selected the top-5 workers who have done most tasks and presented their average accuracy w.r.t. the varying distance in Figure 7 (we divided the distance into 5 ranges, e.g. if the distance is 0.3, it is in range $[0.2, 0.4]$). In general, all workers tended to provide more accurate answers for those close tasks than the distant ones. Also, the impact of distance on different workers varied. For example in the China dataset, when the distance increased from 0.2 to 1.0, w^c_{10} had the best quality and the accuracy of the answers decreased from 90% to 60%, while the accuracy of w^c_{15} decreased from 78% to 45%. This can be attributed to the distance-aware quality of worker in our model: if the worker had better distance-aware quality, the answers had a higher probability to be correct and the impact of distance on it was smaller.

We also investigated the impact of distance on different POIs. To reveal the real influence of POIs, we collected the count of reviews (from Dianping), based on which we categorized the POIs into four classes, as shown in Figure 8. In general, answers on POIs with large influence (i.e. with more reviews) had better accuracy than those POIs with small influence. The accuracy also decreased with the increase of distance. For example in the Beijing dataset, the answers on POIs with the most reviews (#Review>2500) had the best average accuracy. When the distance increased from 0 to 0.4, the impact of distance on it was also minimum as the average accuracy decreased from 90% to 75%. However, for other POIs (e.g., those with less than 500 reviews), the accuracy decreased from 90% to 60%. In our model, the impact of distance on POIs was attributed to the influence of POI: if the POI had a larger influence, its answers had a higher probability to be correct and the impact of distance on it was smaller.

C. Evaluation of the Inference Models

Next, we evaluated the effectiveness of our inference model. To better illustrate the superiority of our model, we first demonstrated a case study.

A Case Study. We consider the labelling task on "Beijing Olympic Forest Park" in the China dataset. The ten answer labels on it were shown in the 1^{st} column of Table I: the labels in bold text (i.e. labels 1,2,3,6,8,10) were the correct labels for this POI, and 4,5,7,9 were the incorrect ones. The inferred results were shown in the 2^{nd} column. As we can see, all the ten labels were accurately inferred. In the 3^{rd}, 4^{th} and 5^{th} columns, we showed the distance and the answers of each of the five workers. In the 6^{th} column, we computed the real accuracy of the five workers' answers based on the ground truth. In the 7^{th} column, we showed the modeled accuracy ($P(z_{t,k}{=}r_{w,t,k})$) of each worker on the task based on our inference model. In the last column, we reported the average accuracy of the 5 workers based on their answers on all tasks. Let us take the 10^{th} label as an example for illustration. Two workers w^c_5, w^c_0 returned "yes" while w^c_{24}, w^c_4 and w^c_{19} returned "no". MV returned incorrect results as the majority voted "no" for this label. It did not consider any worker quality, but in the 6^{th} column, w^c_5 and w^c_0 had much better accuracy than w^c_{24}, w^c_4 and w^c_{19}. EM also returned incorrect results, probably because EM measured the workers' quality based on their average accuracy. As shown in the 8^{th} column, the average accuracy of w^c_{19} and w^c_{24} were higher than w^c_5 and w^c_0, so EM preferred to infer results based on their answers.

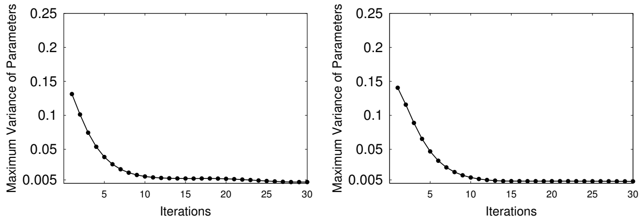


(a) Beijing (b) China

Fig. 9. Accuracy of the Inference Models

Both MV and EM ignored the influence of distance on answer quality, which was captured in our model (IM). For example, w^c_5 and w^c_0 provided high-quality answers as they were much closer to and more familiar with the task. It can be seen in the 7^{th} column that IM provided an estimation accuracy closer to real accuracy than the average accuracy. That may explain why IM had more accurate inference.

Accuracy. We tested the overall accuracy of the three methods by varying the budget from 600 to 1000 and presented the result in Figure 9. We had the following observations: (1) IM outperformed EM and MV across all budgets. For example, when the budget was 1000, IM achieved an overall accuracy of 79%, outperforming EM and MV by 5.2% and 10.1% respectively. This is because MV did not consider the influence of workers' qualities on the results, and EM simply considered an average quality on workers. However, in the POI labelling tasks, as shown in Figures 7 and 8, the distance between POIs and workers had significant impacts on the quality of answers while EM was unaware of it. Our model achieved the best performance as we considered both the influence of workers' inherent qualities and the influence of distances on qualities. (2) With the increase of budget, the accuracy of all methods increased. This is because in a healthy crowdsourcing market we could always receive more correct answers than incorrect answers, which means that we could always get more closely inferred answers towards the real results.



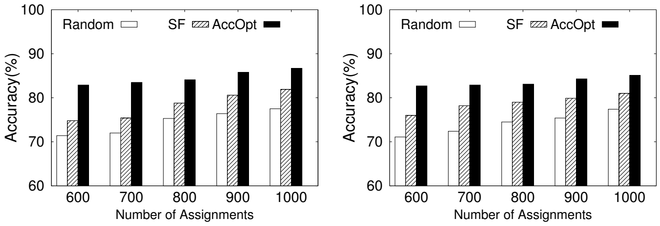
(a) Beijing (b) China

Fig. 10. Convergence of the Inference Models

Convergence. We evaluated the convergence of our model to compute the parameters (see Sec. III-C). We determined the convergence based on the maximum variance of parameters,

TABLE I. A CASE STUDY

POI: Beijing Olympic Forest Park		Assignment and Inference Results							
1. Labels		2. Inferred Result($P(z_{t,k}$=1))		3. Worker	4. Distance	5. Answer	6. Real Accuracy	7. Modeled Accuracy	8. Average Accuracy
[1] park	**[2] Olympics**	[1] 0.99	[2] 0.99	w^c_{24}	0.68	[1,2,3,5,6,7]	60%	59%	63%
[3] sports	[4] the Fragrant Hill	[3] 0.99	[4] 0.14	w^c_5	0.03	[1,2,3,7,8,10]	80%	78%	53%
[5] palace	**[6] stadium**	[5] 0.10	[6] 0.65	w^c_0	0.01	[1,2,3,6,8,9,10]	90%	97%	63%
[7] business	**[8] relax zone**	[7] 0.27	[8] 0.89	w^c_4	0.54	[1,2,4,5,6]	50%	58%	55%
[9] flag-rising	**[10] take a walk**	[9] 0.39	[10] 0.89	w^c_{19}	0.68	[1,2,3,4,5,6]	60%	67%	71%



(a) Beijing (b) China

Fig. 11. Accuracy of Task Assignment Algorithms

TABLE II. EVALUATION OF TASK ASSIGNMENT ALGORITHMS

(a) Beijing

Method	Worker Quality	Percentages of Assigned Workers	Average $Acc_{t,k}$
Random	63.7%	[7%,78%, 15%]	60.2%
SF	68.4%	[22%,55%, 23%]	68.6 %
ACCOPT	69.8%	[8%,77%,15%]	74.5%

(b) China

Method	Worker Quality	Percentages of Assigned Workers	Average $Acc_{t,k}$
Random	65.1%	[7%,82%, 11%]	65.1%
SF	71.6%	[23%,50%, 29%]	70.6 %
ACCOPT	70.1%	[10%,78%,12%]	75.1%

i.e. the maximum difference of parameters from the current iteration to the previous iteration. From Figure 10, we find that the model converged quickly. If we set the convergence threshold as 0.005, the method converged in 23 and 12 iterations for Beijing and China respectively.

D. Evaluation of the Task Assignment Algorithms

Next, we evaluated the accuracy of our proposed task assignment algorithms. The accuracy w.r.t. the varying budget was shown in Figure 11 and more statistics were shown in Table II. We can see that ACCOPT (RANDOM) achieved the best (worst) performance. For example, for 1000 budget, ACCOPT achieved an overall accuracy of 85.1%, outperforming SF and RANDOM by 4.1% and 8.2% respectively. RANDOM had the worst performance because it did not consider workers' qualities when assigning tasks. The 2^{nd} column of Table II recorded the average accuracy for all workers. We find that workers with RANDOM had the worst quality on both datasets. Both SF and ACCOPT optimized the qualities of workers. However, SF optimized the quality by simply considering the distance. In the 3^{rd} column, we divided the tasks into three categories based on the number of assigned workers: less than 3, between 3 and 7, more than 7; then we recorded the percentage of these three categories. Note that the spatial distribution of tasks and workers were not even. For SF, some tasks were assigned to many workers while some were assigned to only a few. For example, in the China dataset, 23% of tasks were only assigned to one or two workers, resulting in only few answers for inference, thus the accuracy of those tasks could not be guaranteed. In the 3^{rd} column of Table II, we tested the average $Acc_{t,k}$ for all the labels and ACCOPT achieved the best value. ACCOPT outperformed SF and RANDOM because it optimized the overall accuracy improvement ($Acc_{t,k}$) for all tasks each time. ACCOPT controlled the assignment through the estimated accuracy, and we can find that the worker quality in the 1^{st} column was optimized and the number of assigned workers in the 2^{nd} column was even.

E. Efficiency and Scalability

First, we reported the average running time of the above inference methods in Figure 12. MV took the least time as it used the simplest inferring strategy. EM and IM had similar elapsed time. For the 1000 number of assignments, IM could converge around 1 second, which is efficient.

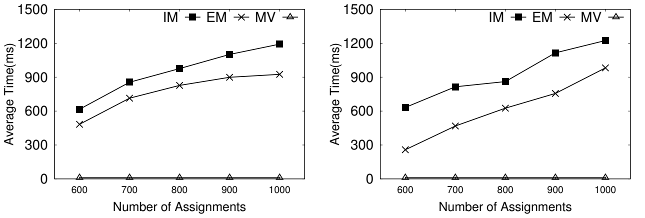
To evaluate the scalability of our approach, we generated a synthetic dataset of POIs and workers, on which we tested the inference model and the task assignment algorithm. First we tested the scalability of our inference model by varying the number of assignments. Figure 13 presented the inference time and the number of iterations. With the increase of number of assignments, we had two observations: (1) The #iterations grew slowly from 29 to 38 as our model can converge quickly. (2) The running time of the parameter estimation increased linearly with the increase of #assignments.

Last we tested the scalability of our task assignment algorithm. In Figure 14(a) we simulated 100 available workers and tested average running time by varying the number of tasks from 2000 to 10000. In Figure 14(b) we used 10000 tasks and varied the number of workers from 20 to 100. We find that our algorithm scaled well and the average running time increased linearly w.r.t. the number of tasks and the number of workers.

VI. RELATED WORKS

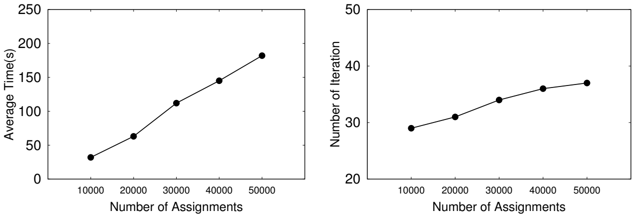
Spatial Crowdsourcing. Crowdsourcing is now becoming a new effective method to handle computer-hard tasks. As many of those tasks contain spatial information (e.g. taking a photo in a location), spatial crowdsourcing also draws attention from both industry and research community [4,13,14,20,21]. A common constraint of those spatial tasks is that, they require workers to finish the tasks by traveling to the marked locations specified in the task. Thus, the spatial distance between workers and tasks is treated as the travel cost, which needs to be considered in the general task objective. Task assignment algorithms are then proposed to optimize those objectives [13,14,20,21] and a platform [4] is developed to specifically support these spatial tasks.

Our work is different from these spatial crowdsourcing tasks. First, the POI labelling task does not request workers to travel to specified locations to answer the task. Second, the optimization goal is different. They focus on minimizing the travel cost while we aim to improve the inference quality. We consider the spatial distance as a factor that can affect the accuracy instead of treating it as a travel cost. Third, they do not consider the accuracy for task assignment at all but treat tasks as finished or unfinished; instead we propose various techniques to optimize the task assignment in term of accuracy.



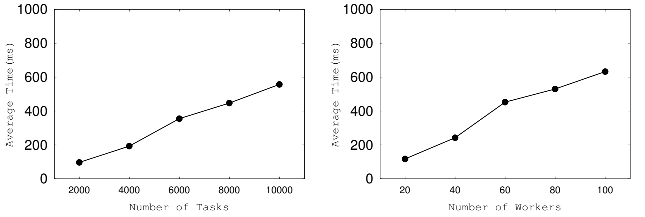
(a) Beijing (b) China

Fig. 12. Elapsed Time of Inference On Real Datasets



(a) Elaspesed Time (b) Number of Iterations

Fig. 13. Scalability of Inference Model



(a) Varying Number of Tasks (b) Varying Number of Workers

Fig. 14. Scalability of Task Assignment

To our best knowledge, this is the first work on how to apply crowdsourcing methods to POI labelling tasks.

Inference Algorithms. Our work is also related to result inferences for crowdsourcing tasks, where different workers give unidentified (e.g. "yes/no") answers and an inference method is required to infer the true result of each task. The general method for result inference is a "voting strategy" which mainly includes two parts: majority voting [3,15] and Bayesian voting [5,12,16,24,26,27]. The majority voting strategy returns the result with the most votings, while the Bayesian voting strategy computes the probability of the result being each answer (e.g. the probability that the result being "yes/no") based on workers' qualities. The Expectation Maximization (EM)-based methods [5,6,12,17,23,24] are the state-of-the-art approaches to estimate the quality of each worker if the ground truth cannot be retrieved. It iteratively updates the workers' qualities and the tasks' true results until convergence. The underlying intuition is that, the workers who usually give correct answers will be measured by a high quality and the answer supported by such high-quality workers will be predicted as a true result. There have been many applications of the EM-based methods in deriving the quality of workers [12,17,24].

The inference model proposed in our work is different from them, as we consider the impact of distance on both workers and POIs in our model for a better quality estimation and result inference. Moreover, we consider the optimal task assignment to improve the overall accuracy based on the inference model.

Task Assignment. Recently, some approaches [2,7,10,11,16, 27] have been proposed to study the task assignment problem. Liu et al. [2,16] adopts an entropy-like method to select the tasks with maximum uncertainty for the worker. Zheng et al. [27] proposes to maximize the evaluation metric-driven quality improvement in the assignment [6]. Fan et al. [7] models diverse accuracies of workers on tasks and assigns tasks to the workers who have high accuracies in answering the tasks. Some other works [10,11] leverage machine learning techniques to decide the assigned tasks under different settings. To summarize, these works only consider how to select the tasks when a single worker comes but neglect the impact of distance to worker quality. In contrast, we consider the optimal task assignment for a set of available workers by considering both the distance-aware quality and the POI influence, and we estimate the accuracy improvement if the task will be assigned to certain workers based on the proposed inference model and then maximize the overall accuracy for all tasks.

VII. Conclusion

In this paper, we studied the crowdsourced POI labelling problem and proposed a framework with an effective label inference model and an online task assigner. In particular, we first proposed a novel model to infer POIs' labels by considering the worker's inherent quality, the worker's distance-aware quality and the influence of POIs to labelling tasks. We proposed an efficient algorithm to compute the parameters in our model. Then we proposed an optimal task assignment algorithm that can judiciously assign tasks to available workers by maximizing the accuracy improvement. Experiment results showed that our approach significantly outperformed state-of-the-art approaches in accuracy and achieved high efficiency.

Acknowledgement

This work was supported by the National Grand Fundamental Research 973 Program of China (2015CB358700), the National Natural Science Foundation of China (61422205, 61373024, 61472198), Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, "NExT Research Center", Singapore (WBS:R-252-300-001-490), Huawei, Shenzhou, FDCT/116/2013/A3, MYRG105(Y1-L3)-FST13-GZ, National 863 Program of China (2012AA012600), Chinese Special Project of Science and Technology (2013zx01039-002-002) and the National Center for International Joint Research on E-Business Information Processing (2013B01035).

References

[1] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW-07*, pages 501–510.

[2] R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W.-C. Tan. Asking the right questions in crowd data sourcing. In *ICDE-12*, pages 1261–1264. IEEE.

[3] C. C. Cao, J. She, Y. Tong, and L. Chen. Whom to ask?: jury selection for decision making tasks on micro-blog services. *VLDB-12*, 5(11):1495–1506.

[4] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang. gmission: a general spatial crowdsourcing platform. *VLDB-14*, 7(13):1629–1632.

[5] A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[6] 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.

[7] J. Fan, G. Li, B. C. Ooi, K.-l. Tan, and J. Feng. icrowd: An adaptive crowdsourcing framework. In *SIGMOD-15*, pages 1015–1030. ACM.

[8] U. P. H. Kellerer and D. Pisinger. Knapsack problems. *Springer*, 2004.

[9] V. Hegde, J. X. Parreira, and M. Hauswirth. Semantic tagging of places based on user interest profiles from online social networks. In *Advances in Information Retrieval*, pages 218–229. Springer, 2013.

[10] C.-J. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *ICML-13*, pages 534–542.

[11] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, volume 12, pages 45–51, 2012.

[12] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[13] L. Kazemi and C. Shahabi. Geocrowd: enabling query answering with spatial crowdsourcing. In *GIS*, pages 189–198. ACM, 2012.

[14] L. Kazemi, C. Shahabi, and L. Chen. Geotrucrowd: trustworthy query answering with spatial crowdsourcing. In *GIS,304-313,2013*.

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

[16] X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. Cdas: a crowdsourcing data analytics system. *VLDB-12*, 5(10):1040–1051.

[17] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller. Human-powered sorts and joins. *VLDB-11*, 5(1):13–24.

[18] R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[19] Y. Song, Z. Zhuang, H. Li, Q. Zhao, J. Li, W.-C. Lee, and C. L. Giles. Real-time automatic tag recommendation. In *SIGIR-08*, pages 515–522.

[20] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *VLDB-14*, 7(10):919–930.

[21] U. ul Hassan and E. Curry. A multi-armed bandit approach to online spatial task assignment. In *UIC*, 2014.

[22] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *VLDB-12*, 5(11):1483–1494.

[23] J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In *SIGMOD*, pages 229–240, 2013.

[24] J. Whitehill, T.-f. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[25] Y. Yu and X. Chen. A survey of point-of-interest recommendation in location-based social networks. In *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[26] Y. Zheng, R. Cheng, S. Maniu, and L. Mo. On optimality of jury selection in crowdsourcing. In *EDBT 2015*, pages 193–204, 2015.

[27] Y. Zheng, J. Wang, G. Li, R. Cheng, and J. Feng. Qasca: A quality-aware task assignment system for crowdsourcing applications. In *SIGMOD-15*, pages 1031–1046. ACM.

Appendix

Proof of Lemma 1. We first consider the case $z_{t,k} \equiv 1$:

$$\begin{aligned}
&P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_1,t,k}, r_{w_2,t,k})\\
&= P(z_{t,k}{=}1|r_{w_1,t,k}{=}1, r_{w_2,t,k}{=}1)P(r_{w_1,t,k}{=}1, r_{w_2,t,k}{=}1)\\
&+ P(z_{t,k}{=}1|r_{w_1,t,k}{=}0, r_{w_2,t,k}{=}1)P(r_{w_0,t,k}{=}0, r_{w_2,t,k}{=}1)\\
&+ P(z_{t,k}{=}1|r_{w_1,t,k}{=}1, r_{w_1,t,k}{=}0)P(r_{w_1,t,k}{=}1, r_{w_2,t,k}{=}0)\\
&+ P(z_{t,k}{=}1|r_{w_1,t,k}{=}0, r_{w_2,t,k}{=}0)P(r_{w_1,t,k}{=}0, r_{w_2,t,k}{=}0).
\end{aligned}$$

Since $r_{w_1,t,k}$ and $r_{w_2,t,k}$ are independent, we have $P(r_{w_1,t,k}, r_{w_2,t,k}){=}P(r_{w_1,t,k})P(r_{w_2,t,k}){=}P(r_{w_2,t,k}, r_{w_1,t,k})$, thus $P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_1,t,k}, r_{w_2,t,k}){=}P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_2,t,k}, r_{w_1,t,k})$. Similarly, for the case $z_{t,k} \equiv 0$, $P_{\mathbb{E}}(z_{t,k} = 0|r_{w_1,t,k}, r_{w_2,t,k}) = P_{\mathbb{E}}(z_{t,k} = 0|r_{w_2,t,k}, r_{w_1,t,k})$. To conclude, $Acc_{t,k}(w_1, w_2) = Acc_{t,k}(w_2, w_1)$.

Proof of Lemma 2. To simplify the proof, we first prove that Lemma 2 holds for the simple situation when $\widehat{W}(t){=}\{w_1, w_2\}$.

According to the proof of Lemma 1, for the case $z_{t,k} \equiv 1$:

$$\begin{aligned}
&P(z_{t,k}{=}1|r_{w_1,t,k}{=}1, r_{w_2,t,k}{=}1)P(r_{w_1,t,k}{=}1)\\
&+P(z_{t,k}{=}1|r_{w_1,t,k}{=}0, r_{w_2,t,k}{=}1)P(r_{w_1,t,k}{=}0)\\
&{=}\frac{|W(t)|P(z_{t,k}{=}1){+}P(z_{t,k}{=}r_{w_1,t,k}){+}P(z_{t,k}{=}r_{w_2,t,k})}{|W(t)|{+}2}P(z_{t,k}{=}r_{w_1,t,k})\\
&{+}\frac{|W(t)|P(z_{t,k}{=}1){+}P(z_{t,k}{\neq}r_{w_1,t,k}){+}P(z_{t,k}{\neq}r_{w_2,t,k})}{|W(t)|{+}2}P(z_{t,k}{\neq}r_{w_1,t,k})\\
&{=}\frac{|W(t)|{+}1}{|W(t)|{+}2}(\frac{|W(t)|P(z_{t,k}{=}1){+}P(z_{t,k}{=}r_{w_1,t,k})}{|W(t)|{+}1}P(z_{t,k}{=}r_{w_1,t,k})\\
&{+}\frac{|W(t)|P(z_{t,k}{=}1){+}P(z_{t,k}{\neq}r_{w_1,t,k})}{|W(t)|{+}1}P(z_{t,k}{\neq}r_{w_1,t,k}){+}\frac{P(z_{t,k}{=}r_{w_2,t,k})}{|W(t)|{+}1})\\
&{=}\frac{|W(t)|{+}1}{|W(t)|{+}2}(P_{\mathbb{E}}(z_{t,k} = 1|r_{w_1,t,k}){+}\frac{P(z_{t,k}{=}r_{w_2,t,k})}{|W(t)| + 1})\\
&{=}\frac{(|W(t)| + 1)P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_1,t,k}){+}P(z_{t,k}{=}r_{w_2,t,k})}{|W(t){+}2|}.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&P(z_{t,k}{=}1|r_{w_1,t,k}{=}1, r_{w_2,t,k}{=}0)P(r_{w_1,t,k}{=}1)\\
&+P(z_{t,k}{=}1|r_{w_1,t,k}{=}0, r_{w_2,t,k}{=}0)P(r_{w_1,t,k}{=}0)\\
&{=}\frac{(|W(t)|{+}1)(P_{\mathbb{E}}(z_{t,k} = 1|r_{w_1,t,k}){+}P(z_{t,k}{\neq}r_{w_2,t,k})}{|W(t){+}2|}.
\end{aligned}$$

To this end, $P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_1,t,k}, r_{w_2,t,k})$

$$\begin{aligned}
&{=}\frac{(|W(t)|{+}1)P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_1,t,k}){+}P(z_{t,k}{=}r_{w_2,t,k})}{|W(t){+}2|}P(z_{t,k}{=}r_{w_2,t,k})\\
&{+}\frac{(|W(t)|{+}1)P_{\mathbb{E}}(z_{t,k}{=}1|r_{w_1,t,k}){+}P(z_{t,k}{\neq}r_{w_2,t,k})}{|W(t){+}2|}P(z_{t,k}{\neq}r_{w_2,t,k}).
\end{aligned}$$

Similarly, the equation still holds for the case $z_{t,k} \equiv 0$. By aggregating the cases for $z_{t,k} \equiv 1$ and $z_{t,k} \equiv 0$, Lemma 2 holds for $\widehat{W}(t) = \{w_1, w_2\}$.

In general, we can divide an arbitrary $\widehat{W}(t)$ into w and $\widehat{W}(t) - \{w\}$. By replacing $\widehat{W}(t) - \{w\}$ as w_1 and w as w_2 into the above proof, we can prove that the equation holds for any $\widehat{W}(t)$.

Thus, we can prove the lemma.

Proof of Lemma 3. We prove the NP-hardness by a reduction from the n-th order knapsack problem (nOKP) [3,8]. An nOKP is a Knapsack problem aims to maximize:

$$\sum_{i_1}\sum_{i_2}\cdots\sum_{i_n} V[i_1, i_2, \cdots, i_n]) \cdot x_1 x_2 \cdots x_n$$

where $V[i_1, i_2, \cdots, i_n]$ is the profit achieved if item $x_1, x_2, \cdots x_n$ are selected into the knapsack simultaneously. Consider an instance of optimal task assignment problem with $h = 1$. The problem is equivalent to selecting $n = |W|$ items into the knapsack (where each worker is assigned with one task) from $|W| \cdot |T|$ tasks simultaneously and the profit is the overall expected accuracy.

Thus the Optimal Task Assignment Problem is NP-hard.