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 new 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.

1. 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
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 com-
ponents: (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 assign-
ment (Section II).

(2) We develop an effective inference model which uti-
lizes 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 crowdsourc-
ing platform, and the results show that our methods signifi-
cantly 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 \( t = \{O_{ti},L_{ti}\} \) includes a POI
\( O_{ti} \) (with a name and a geo-location) and a label set \( L_{ti} = \{l_{ti,1},l_{ti,2},\ldots,l_{ti,|L_{ti}|}\} \). For simplicity of presentation, in the rest
of the paper we use task \( t \) and POI \( O_{ti} \) 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 \( l_{ti,k} \) (\( 1 \leq k \leq |L_{ti}| \)) has a true result 1/0 (“yes/no”), where
1 (0) indicates that \( l_{ti,k} \) is a correct (an incorrect) label for \( O_{ti} \).

Workers. Each worker \( w \) has a location (e.g. home, office).
For each task \( t = \{O_{ti},L_{ti}\} \), workers are asked to select labels
from \( L_{ti} \) which they think correct for \( O_{ti} \). We denote the answer
set by \( R = \{(w,t,\mathcal{R}(w,t))\} \), where \( \mathcal{R}(w,t) = \{r_{w,t,k} \mid 1 \leq k \leq |L_{ti}| \} \) is \( w \’s \) answer for a task \( t \) and \( r_{w,t,k} = 1/0 \) is \( w \’s \) answer for a certain label \( l_{ti,k} \). 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 \( r_{w,t,1} = 0 \).

Task Assignment. When a set 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 \( \mathcal{A}(W) = \{A(w)\mid w \in W\} \) where \( A(w) \) is the set of \( h \)
tasks assigned to worker \( w \). A worker can do multiple tasks and
we denote \( T(w) = \{t \in T\mid t \text{ is done by } w\} \) as the set of tasks
already done by worker \( w \). Meanwhile, a task can be answered
by several workers, we denote \( W(t) = \{w \mid w \text{ has done task } t\} \)
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.
\[
\text{accuracy} = \frac{1}{|T|} \sum_{t_i \in T} \frac{N_{t_i}^i}{|L_{ti}|},
\]
where \( N_{t_i}^i \) is the number of labels that an algorithm accurately
reports for task \( t_i \). For example, assuming \(|L_{ti}|=10\) for task
\( t_i \) and the first 3 labels are the true labels; if an algorithm
identifies the 1st and 4th label as the correct ones, then \( N_{t_i}^i=7 \).

There are two sub-problems to study in achieving a high-
quality crowdsourced POI labelling. (1) The result inference
problem: given the answer set \( R \) 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 \( R \);
(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 \( R \) 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 $P(z_{t,k}) \geq 0.5$, 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

$$WQ_w = P(i_w).$$

A higher $WQ_w$ 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 $t$ ($0 \leq d(w, t) \leq 1$). $d(w, t)$ 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.

**Definition 3 (Bell-Shaped Function):**

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

where $\lambda$ is a parameter to control the decrease degree of the function value with the increase of distance $d(w, t)$. If $\lambda$ is large, then the quality decreases quickly with the increase of distance. For example, Figure 4 shows three functions with $\lambda$ 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 $\lambda$.

However, simply using a single distance function with an unknown parameter $\lambda$ 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 $\lambda$ in a probability model, because there is no closed-form solution to compute $\lambda$ 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 $F$ consists of a set of bell-shaped functions with fixed parameters $\lambda_1, \lambda_2, \ldots, \lambda_{|F|}$, i.e.,

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

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

**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.,

$$d(w, t) = \max_{\lambda \in F} f_{\lambda}(d(w, t)).$$

$\lambda$ represents the 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.
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 \( f_\lambda \) 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 \( P(d_w = f_{100}) = 0.6, P(d_w = f_{10}) = 0.2, P(d_w = f_{01}) = 0.2, \) then \( w \) can possibly provide accurate answers only for nearby POIs. Otherwise, if \( P(d_w = f_{100}) = 0.2, P(d_w = f_{10}) = 0.2, P(d_w = f_{01}) = 0.6, \) then \( w \) is able to provide accurate answers for distant POIs. In Figure 4, if \( P(d_w = f_{100}) = P(d_w = f_{10}) = P(d_w = f_{01}) = \frac{1}{3}, \) 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 \( IQ_t \) as a combination of distance functions

\[
IQ_t = \sum_{f_\lambda \in F} P(d_t = f_\lambda) \cdot f_\lambda(d(w, t)).
\]

where \( d_t \) is also a random variable with multinomial distribution over the set and \( P(d_t = f_\lambda) \) is the weight of \( f_\lambda \) in the function set for \( d_t \). For the distance function set in Figure 4, if a POI has a large influence, then \( P(d_t = f_{01}) \) 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

\[
P(r_{w,t,k} = z_{t,k} \mid i_w = 0) = 0.5.
\]

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:

\[
P(r_{w,t,k} = z_{t,k} \mid i_w = 1) = \alpha \cdot \sum_{f_\lambda \in F} P(d_w = f_\lambda) \cdot f_\lambda(d(w, t)) + (1 - \alpha) \cdot \sum_{f_\lambda \in F} P(d_t = f_\lambda) \cdot f_\lambda(d(w, t)),
\]

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

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
W & R(w, t) & P(i_w = 1) & P(d_w) & T & P(z_{t,k} = 1) & P(d_t) \\
\hline
w_1 & t_1:[1,1,0], t_2:[1,0,0] & 0.89 & 0.07 & 0.12 & 0.81 & 0.60 & 64.0 & 64.0 & 0.35 & 0.60 & 64.0 & 0.69 \\
\hline
w_2 & t_2:[1,1,0], t_3:[1,1,0] & 0.93 & 0.04 & 0.09 & 0.87 & 0.72 & 72.0 & 0.25 & 0.04 & 0.06 & 0.90 \\
\hline
w_3 & t_2:[1,1,0], t_4:[1,0,0] & 0.93 & 0.05 & 0.06 & 0.89 & 0.71 & 49.0 & 0.28 & 0.06 & 0.07 & 0.87 \\
\hline
w_4 & t_2:[0,0,0], t_4:[1,1,1] & 0.19 & 0.41 & 0.40 & 0.19 & 0.59 & 0.40 & 0.40 & 0.24 & 0.24 & 0.52 \\
\hline
\end{array}
\]

**Fig. 3. A Running Example**

\[
DQ_w = \sum_{f_\lambda \in F} P(d_w = f_\lambda) \cdot f_\lambda(d(w, t))
\]

\[
IQ_t = \sum_{f_\lambda \in F} P(d_t = f_\lambda) \cdot f_\lambda(d(w, t)).
\]

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 \( w \):
  - 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 \( z_{t,k}, i_w, d_w, d_t \). In fact, it is a simple deduction of the modeled answer accuracy introduced in Equation 9:

\[
P(r_{w,t,k} = z_{t,k}) = P(r_{w,t,k} = z_{t,k} | z_{t,k} = 1) \cdot P(z_{t,k} = 1)
\]

\[
+ P(r_{w,t,k} = z_{t,k} | z_{t,k} = 0) \cdot P(z_{t,k} = 0).
\]

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 \( t_4 \). The inference results of \( t_1, t_2, t_3 \) 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 \( P(z_{t,k} = r_{w_2,t_4,k}) = 0.87 \).

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

\[
\arg\max_{z_{t,k}, i_w, d_w, d_t} \prod_{t,w,l} P(r_{w,t,k})
\]

\[
= \prod_{t,w,l} \sum_{z_{t,k}, i_w, d_w, d_t} P(r_{w,t,k})P(z_{t,k}, i_w, d_w, d_t)P(i_w)P(z_{t,k})P(d_w)P(d_t)
\]

(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

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

\[
\propto P(r_{w,t,k} = z_{t,k} = 1 | i_w = 0)P(z_{t,k})P(i_w)P(d_w)P(d_t)
\]

Case 2: For \( i_w = 0 \) and \( z_{t,k} = 1 \), we have

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

\[
\propto P(r_{w,t,k} = z_{t,k} = 0 | i_w = 0)P(z_{t,k})P(i_w)P(d_w)P(d_t)
\]

Case 3: For \( i_w = 1 \) and \( z_{t,k} = 0 \), we have

\[
q(d_w, d_t) = \alpha f_{d_w}(d(w,t)) + (1 - \alpha)f_{d_t}(d(w,t))
\]

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

\[
=q(d_w, d_t)^{1-r_{w,t,k}} (1-q(d_w, d_t))^{r_{w,t,k}} P(z_{t,k})P(i_w)P(d_w)P(d_t)
\]

Case 4: For \( i_w = 1 \) and \( z_{t,k} = 1 \), we have

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

\[
=q(d_w, d_t)^{1-r_{w,t,k}} (1-q(d_w, d_t))^{r_{w,t,k}} P(z_{t,k})P(i_w)P(d_w)P(d_t)
\]

(12)

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

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

(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

\[
P(z_{t,k}) = \sum_{w \in W(t)} \sum_{i_w, d_w, d_t} P(z_{t,k}, i_w, d_w, d_t | r_{w,t,k}) | W(t) |
\]

\[
P(i_w) = \sum_{t \in T(w)} \sum_{z_{t,k}, d_w, d_t} \sum_{r_{w,t,k}} P(z_{t,k}, i_w, d_w, d_t | r_{w,t,k}) | L_t |
\]

\[
P(d_w) = \sum_{t \in T(w)} \sum_{z_{t,k}, i_w, d_t} \sum_{r_{w,t,k}} P(z_{t,k}, i_w, d_w, d_t | r_{w,t,k}) | L_t |
\]

\[
P(d_t) = \sum_{t \in T(w)} \sum_{z_{t,k}, i_w, d_w} \sum_{r_{w,t,k}} P(z_{t,k}, i_w, d_w, d_t | r_{w,t,k}) | L_t |
\]

(14)

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., \( z_{t,k} = 0 \)), the accuracy of our inference is \( P(z_{t,k} = 0) \).
Formally we denote $\text{Acc}_{t,k}$ as the accuracy of our inference on a label $l_{t,k}$, where

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

In task assignment, if the task $t$ is assigned to a set of new workers, denoted by $\hat{W}(t) \subseteq W$, $\text{Acc}_{t,k}$ will change as the answer set of $t$ changes. Let $\text{Acc}_{t,k}(\hat{W}(t))$ denote the accuracy of $t$ after workers in $\hat{W}(t)$ submit their answers. Intuitively, we will assign $t$ to workers in $\hat{W}(t)$ if the accuracy improvement (i.e., $\Delta \text{Acc}_{t,k}(\hat{W}(t)) = \text{Acc}_{t,k}(\hat{W}(t)) - \text{Acc}_{t,k}$) is large. Thus we need to know how much quantity $\text{Acc}_{t,k}(\hat{W}(t))$ will change if $t$ is assigned to $\hat{W}(t)$. To address this problem, we first discuss how to estimate the accuracy $\text{Acc}_{t,k}(\hat{W}(t))$ in Section IV-B. As the accuracy $\text{Acc}_{t,k}(\hat{W}(t))$ 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., $\hat{W}(t) = \{w\}$. Let $\text{Acc}_{t,k}(w)$ 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 $\text{Acc}_{t,k}(w)$.

Since $\text{Acc}_{t,k}(w)$ 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 $\text{Acc}_{t,k}(w) = P(z_{t,k} = 1) = P(r_{w,t,k})$, where $P(z_{t,k} = 1) = P(r_{w,t,k})$ 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

$$\begin{align*}
P(z_{t,k} = 1|r_{w,t,k}) &= \sum_{w \in \hat{W}(t)} \sum_{i_w,d_w,d_t} P(z_{t,k} = 1, i_w, d_w, d_t | r_{w,t,k}) \\
&= \frac{\sum_{i_w,d_w,d_t} P(z_{t,k} = 1, i_w, d_w, d_t | r_{w,t,k})}{\sum_{i_w,d_w,d_t} P(z_{t,k} = 1, i_w, d_w, d_t | r_{w,t,k})} \\
&= \frac{|\hat{W}(t)| + 1}{|\hat{W}(t)| + 1} \\
&= \frac{|\hat{W}(t)| + 1}{|\hat{W}(t)| + 1}.
\end{align*}$$

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 93.

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 $P(z_{t,k} = 1), P(r_{w,t,k} = 1) = P(r_{w,t,k} = z_{t,k})$ and $P(r_{w,t,k} = 0) = P(r_{w,t,k} \neq z_{t,k})$, thus we have

$$\begin{align*}
P(z_{t,k} = 1|r_{w,t,k}) &= P(z_{t,k} = 1 | r_{w,t,k} = 1) P(r_{w,t,k} = 1) + P(z_{t,k} = 1 | r_{w,t,k} = 0) P(r_{w,t,k} = 0) \\
&= \frac{|\hat{W}(t)| P(z_{t,k} = 1) + P(z_{t,k} = r_{w,t,k})}{|\hat{W}(t)| + 1}.
\end{align*}$$

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

$$\begin{align*}
P(z_{t,k} = 0|r_{w,t,k}) &= \frac{|\hat{W}(t)| P(z_{t,k} = 0) + P(z_{t,k} = r_{w,t,k})}{|\hat{W}(t)| + 1}.
\end{align*}$$

To conclude, we compute $\text{Acc}_{t,k}(w)$ 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:

$$\text{Acc}_{t,k}(w) = \left\{ \begin{array}{ll}
P(z_{t,k} = 1 | r_{w,t,k}) & (z_{t,k} \equiv 1) \\
P(z_{t,k} = 0 | r_{w,t,k}) & (z_{t,k} \equiv 0) \end{array} \right.$$ (18)

#### Example 2

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

#### 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 $\text{Acc}_{t,k}(w_1, w_2)$ denote the accuracy of $l_{t,k}$ after $w_1$ and $w_2$ give answers to $l_{t,k}$. We have $\text{Acc}_{t,k}(w_1, w_2) = \text{Acc}_{t,k}(w_2, w_1)$

**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 $\hat{W}(t)$, we do not need to consider the sequence of these workers. Next we introduce how to estimate $\text{Acc}_{t,k}(\hat{W}(t))$.

Recall that $\text{Acc}_{t,k}(\hat{W}(t))$ is an expected probability based on the value of $r_{w,t,k}$ ($w \in \hat{W}$). To compute $\text{Acc}_{t,k}(\hat{W}(t))$, we have to enumerate all possible combinations of $r_{w,t,k}$.
with $O(2^{\hat{W}(t)})$ time. Fortunately we find that $Acc_{t,k}(\hat{W}(t))$ satisfies the following recursive property (Lemma 2) and thereby $Acc_{t,k}(\hat{W}(t))$ can be calculated in linear time.

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

$$
Acc_{t,k}(\hat{W}(t)) = \begin{cases} 
    P_{E}(z_{t,k} = 1|\hat{W}(t)) & (z_{t,k} \equiv 1) \\
    P_{E}(z_{t,k} = 0|\hat{W}(t)) & (z_{t,k} \equiv 0) 
\end{cases}
$$

$$
= (|W(t)| + |\hat{W}(t)| - 1) \cdot Acc(\hat{W}(t) - |w|)P(z_{t,k} = r_{w,t,k})P(z_{t,k} = r_{w,t,k})
+ (|W(t)| + |\hat{W}(t)| - 1) \cdot Acc(\hat{W}(t) - |w|)P(z_{t,k} = r_{w,t,k})P(z_{t,k} = r_{w,t,k}),
$$

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

**Proof:** See appendix. 

Based on Lemma 2, we can estimate $Acc_{t,k}(\hat{W}(t))$ in linear time $O(\hat{W}(t))$. Suppose $\hat{W}(t) = \{w_1, w_2, \ldots, w_4\}$, we can first compute $Acc_{t,k}(w_1)$, then compute $Acc_{t,k}(w_1, w_2)$ based on $Acc_{t,k}(w_1) \in O(1)$ time and repeat the computation until we get $Acc_{t,k}(w_1, w_2, \ldots, w_4)$. 

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

**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

$$
\Delta Acc_{t,k}(\hat{W}(t)) = \mathbb{E}[z_{t,k} = 1] \cdot (P(z_{t,k} = 1|\hat{W}(t)) - P(z_{t,k} = 1)) + \mathbb{E}[z_{t,k} = 0] \cdot (P(z_{t,k} = 0|\hat{W}(t)) - P(z_{t,k} = 0)).
$$

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.,

$$
\arg\max_{\mathcal{A}(w), w \in W} \sum_{t \in T} \sum_{k = 1} \Delta Acc_{t,k}(\hat{W}(t))
$$

s.t. $|\mathcal{A}(w)| = h$

where $\mathcal{A}(w)$ is the set of $h$ tasks assigned to worker $w$, and $\hat{W}(t) = \{w|t \in \mathcal{A}(w)\}$ 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. $\mathcal{A}(W) = \{\mathcal{A}(w) = \emptyset | w \in W\}$
2. foreach $t \in T$ do
   3. $\hat{W}(t) = \emptyset$
   4. foreach $w \in W$ do
      5. foreach $t \in T$ do
         6. for $k = 1$ to $|I_t|$ do
            7. $\Delta Acc[w][t[k] = Acc_{t,k}(w)$;
            8. $\Delta Acc_{t,k}(\hat{W}(t) \cup \{w\})$
       9. while $|\mathcal{A}(W)| < h \cdot |W|$ do
          10. $t_{max}, w_{max} = \arg \max_{t_{max}, w_{max}} \Delta Acc$
          11. $\mathcal{A}(w_{max}), \text{append}(t_{max})$
          12. $\hat{W}(t_{max}), \text{append}(w_{max})$
          13. $\Delta Acc_{t,k}(\hat{W}(t_{max}))$
          14. if $|\mathcal{A}(w_{max})| > h$ then
             15. $\Delta Acc_{t,k}(\hat{W}(t_{max}))$
          16. foreach $w \in W - \hat{W}(t_{max})$ do
             17. for $k = 1$ to $|I_t|$ do
                18. $\Delta Acc[w][t[k] = Acc_{t,k}(\hat{W}(t) \cup \{w\})$
                19. $\Delta Acc_{t,k}(\hat{W}(t_{max}))$
          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 $\hat{W}(t)$ as empty set (lines 1-3). Then it computes all the estimated accuracy and keeps them in a matrix $Acc$, where $Acc[w][t][k] = Acc_{t,k}(w)$ computed in Equation 18 (line 7). Notice that in practice, $Acc[w][t][k]$ stores a pair of values $(P_E(z_{t,k} = 1|r_{w,t,k}), P_E(z_{t,k} = 0|r_{w,t,k}))$. Meanwhile, it also initializes a matrix $\Delta Acc$ to record all the accuracy improvement, where $\Delta Acc[w][t]$ is the accuracy improvement $\Delta Acc_{t,k}(\hat{W}(t))$ 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 $\Delta Acc$ and put them into $\mathcal{A}(W)$ and $\hat{W}(t)$ (lines 10-13). If $h$ tasks have been assigned to worker $w_{max}$, it removes $w_{max}$ from the matrix $\Delta Acc$ (line 15) to avoid duplicate assignments. As workers in $\hat{W}(t_{max})$ have been assigned with $t_{max}$, for the rest workers in $W - \hat{W}(t_{max})$, we update matrix
Acc and $\Delta$Acc for $t_{\text{max}}$ by assuming that $w$ is further assigned with $t_{\text{max}}$ (line 18-19). The algorithm terminates when all workers have been assigned with $h$ tasks.

**Time Complexity.** To initialize the matrices of Acc and $\Delta$Acc, 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 $\Delta$Acc of $t_{\text{max}}$ 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 Fig. 3. Suppose $W = \{w_2\}$ and $h = 1$, for label $t_{4,1}$, we have computed $P_{b}(z_{4,1} = 1|r_{w_2,t_{4,1}}) = 0.65$ and $P_{b}(z_{4,1} = 0|r_{w_2,t_{4,1}}) = 0.53$ in Example 2. Therefore, its expected accuracy improvement is $\Delta \text{Acc}_{4,1}(w_2) = 0.59 \times (0.65 - 0.59) + 0.41 \times (0.53 - 0.41) = 0.08$. Similarly, we can compute $\Delta \text{Acc}_{4,2}(w_2) = \Delta \text{Acc}_{4,3}(w_2) = 0.08$. 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 Dianping4. 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 ChinaCrowds5, 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.

---

4http://www.dianping.com/
5http://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, $w_{10}^C$ had the best quality and the accuracy of the answers decreased from 90% to 60%, while the accuracy of $w_{10}^C$ 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 1st 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 2nd column. As we can see, all the ten labels were accurately inferred. In the 3rd, 4th and 5th columns, we showed the distance and the answers of each of the five workers. In the 6th column, we computed the real accuracy of the five workers’ answers based on the ground truth. In the 7th column, we showed the modeled accuracy ($P(z_t=k|w_{t,k})$) 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 10th label as an example for illustration. Two workers $w_5^G$ and $w_6^G$ returned “yes” while $w_2^C$, $w_4^C$ and $w_0^G$ returned “no”. MV returned incorrect results as the majority voted “no” for this label. It did not consider any worker quality, but in the 6th column, $w_5^G$ and $w_6^G$ had much better accuracy than $w_2^C$, $w_4^C$ and $w_0^G$. EM also returned incorrect results, probably because EM measured the workers’ quality based on their average accuracy. As shown in the 8th column, the average accuracy of $w_0^G$ was higher than $w_5^G$ and $w_6^G$, so EM preferred to infer results based on their answers.

Both MV and EM ignored the influence of distance on answer quality, which was captured in our model (IM). For example, $w_5^G$ and $w_6^G$ provided high-quality answers as they were much closer to and more familiar with the task. It can be seen in the 7th 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 IM 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.

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,
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 2nd 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 3rd 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 3rd column of Table II, we tested the average ACCOPT for all the labels and ACCOPT achieved the best value. ACCOPT outperformed SF and RANDOM because it optimized the overall accuracy improvement (ACCt,k) for all tasks each time. ACCOPT controlled the assignment through the estimated accuracy, and we can find that the worker quality in the 1st column was optimized and the number of assigned workers in the 2nd 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.
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.

VIII. 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, Shenzhen, 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

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

$$P_E(z_{t,k}=1| r_{w_1,t,k}, r_{w_2,t,k}) = P(z_{t,k}=1| r_{w_1,t,k}=1, r_{w_2,t,k}=1) P(r_{w_1,t,k}=1, r_{w_2,t,k}=1)$$

$$+ P(z_{t,k}=1| r_{w_1,t,k}=0, r_{w_2,t,k}=1) P(r_{w_1,t,k}=0, r_{w_2,t,k}=1)$$

$$+ P(z_{t,k}=1| r_{w_1,t,k}=1, r_{w_1,t,k}=0) P(r_{w_1,t,k}=1, r_{w_2,t,k}=0)$$

$$+ P(z_{t,k}=1| r_{w_1,t,k}=0, r_{w_2,t,k}=0) P(r_{w_1,t,k}=0, r_{w_2,t,k}=0).$$

Since $r_{w_1,t,k}$ and $r_{w_2,t,k}$ are independent, we have

$$P(r_{w_1,t,k}, r_{w_2,t,k}) = P(r_{w_1,t,k}) P(r_{w_2,t,k}) = P(r_{w_1,t,k}=k, r_{w_2,t,k}=k).$$

Thus,

$$P_E(z_{t,k}=1| r_{w_1,t,k}, r_{w_2,t,k}) = P_E(z_{t,k}=1| r_{w_1,t,k}, r_{w_2,t,k}).$$

Similarly, for the case $z_{t,k} \equiv 0$,

$$P_E(z_{t,k}=0| r_{w_1,t,k}, r_{w_2,t,k}) = P_E(z_{t,k}=0| r_{w_1,t,k}, r_{w_2,t,k}).$$

To conclude,

$$A_{t,k}(w_1, w_2) = A_{t,k}(w_1, w_2).$$

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

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

$$P(z_{t,k}=1| r_{w_1,t,k}=1, r_{w_2,t,k}=1) P(r_{w_1,t,k}=1, r_{w_2,t,k}=1)$$

$$+ P(z_{t,k}=1| r_{w_1,t,k}=0, r_{w_2,t,k}=1) P(r_{w_1,t,k}=0, r_{w_2,t,k}=1)$$

$$= \frac{|W(t)| P(z_{t,k}=1)+ P(z_{t,k}=w_{w_1,t,k}+ P(z_{t,k}=w_{w_2,t,k})}{|W(t)|} + 2$$

$$+ \frac{|W(t)| P(z_{t,k}=1)+ P(z_{t,k}=w_{w_1,t,k}+ P(z_{t,k}=w_{w_2,t,k})}{|W(t)|} + 2$$

Similarly, we have

$$P(z_{t,k}=1| r_{w_1,t,k}=1, r_{w_2,t,k}=0) P(r_{w_1,t,k}=1, r_{w_2,t,k}=0)$$

$$+ P(z_{t,k}=1| r_{w_1,t,k}=0, r_{w_2,t,k}=0) P(r_{w_1,t,k}=0, r_{w_2,t,k}=0)$$

$$= \frac{|W(t)|+1}{|W(t)|+2} P(z_{t,k}=1| r_{w_1,t,k}) + P(z_{t,k}=r_{w_2,t,k})$$

$$= \frac{|W(t)|+1}{|W(t)|+2} (P_E(z_{t,k}=1| r_{w_1,t,k}) + P(z_{t,k}=r_{w_2,t,k})$$

$$= \frac{|W(t)|+1}{|W(t)|+2} (P_E(z_{t,k}=1| r_{w_1,t,k}) + P(z_{t,k}=r_{w_2,t,k})$$

To this end, $P_E(z_{t,k}=1| r_{w_1,t,k}=1, r_{w_2,t,k})$

$$= \frac{|W(t)|+1}{|W(t)|+2} P_E(z_{t,k}=1| r_{w_1,t,k}) + P(z_{t,k}=r_{w_2,t,k})$$

$$= \frac{|W(t)|+1}{|W(t)|+2} (P_E(z_{t,k}=1| r_{w_1,t,k}) + P(z_{t,k}=r_{w_2,t,k})$$

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 $\hat{W}(t) = \{w_1, w_2\}$.

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

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:

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

where $V[i_1, i_2, \ldots, i_n]$ is the profit achieved if item $x_1, x_2, \ldots, x_n$ are selected into the knapsack (where each item 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.