Towards Personalized Task Matching in Mobile Crowdsensing via Fine-Grained User Profiling

Shuo Yang†, Kunyan Han†, Zhenzhe Zheng†, Shaojie Tang‡, Fan Wu†
†Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
‡Department of Information Systems, University of Texas at Dallas, USA
†{wnmmxy, hankunyan, zhengzhenzhe}@sjtu.edu.cn; ‡tangshaojie@gmail.com; †fwu@cs.sjtu.edu.cn

***Abstract*—In mobile crowdsensing, finding the best match between tasks and users is crucial to ensure both the quality and effectiveness of a crowdsensing system. Existing works usually assume a centralized task assignment by the platform, without addressing the need of fine-grained personalized task matching. In this paper, we argue that it is essential to match tasks to users based on a careful characterization of both the users' preferences and reliability levels. To that end, we propose a personalized task recommender system for mobile crowdsensing, which recommends tasks to users based on a recommendation score that jointly takes each user's preference and reliability into consideration. We first present a simple but effective method to profile the users' preferences by exploiting the implicit feedback from their historical performance. Then, to profile the users' reliability levels, we formalize the problem as a semi-supervised learning model, and propose an efficient block coordinate descent algorithm to solve the problem. For some tasks that lack historical information, we further propose a matrix factorization method to infer the users' reliability on those tasks. We conduct extensive experiments to evaluate the performance of our system, and the evaluation results demonstrate that our system can achieve superior performance to our benchmarks in both user profiling and personalized task matching.**

I. Introduction

Due to the rapid development of smart devices and wireless technology, mobile crowdsensing [1] has risen as an emerging sensing paradigm. It can employ a large number of smart devices to extract and share their local information using their embedded sensors. A typical mobile crowdsensing system usually consists of three major components: crowdsensing platform, service requesters, and mobile device users. The platform is responsible for handling information requests from the service requesters and publishing sensing tasks to the users through the interaction of their smartphone applications.

A critical problem in crowdsensing is to find the best match between users and tasks. Most of the existing works adopt a *platform-centric model* [2]–[7], which allows the platform to make centralized decisions on which users are selected to perform which sensing tasks. These works usually focus on the incentive problem, where a typical procedure goes like this: each user submits a bid reflecting her willingness or cost in participating in a task, and then the platform determines the set of selected users and their payments, so as to optimize certain utility metric and satisfy some game-theoretic properties. The underlying assumption behind this type of model is that the users are fully rational and are capable of determining their optimal strategies. However, as pointed out in [8], this assumption, as well as the setting that each user's preference can be abstracted as a single bidding parameter, could be an oversimplification of the complicated user behaviors.

Another type of task matching systems, referred as *user-centric model*, gives the users more freedom to choose their interested tasks. It has been widely adopted in many commercial crowdsensing systems, such as Waze [9], Field Agent [10], and Gigwalk [11]. In these systems, the available tasks are shown to the users via their smartphone applications. The users can manually browse through the task corpus (often with simple built-in filters, such as proximity filter and payment filter), and choose their interested tasks to participate in. However, since the number of tasks is often really large, it is inefficient for the users to browse page by page searching for suitable tasks. Without an efficient personalized task matching solution, the users may end up selecting tasks that they are not familiar with or not interested in, which may result in a decrement of the quality of their collected sensing data.

Considering the limitations of existing task matching works, we propose to design a personalized task recommender system for crowdsensing, so as to facilitate the match of the users with suitable tasks. We note that in traditional recommender systems, such as movie recommendation, items are recommended based only on customers' preferences [12]. Whereas, in mobile crowdsensing, besides the metric of the users' preferences, we also need to take the users' reliability/data quality into consideration. That is because the users may have heterogeneous sensing behaviors towards different tasks, which could influence the quality of their collected data [13]. Achieving preference- and quality-aware task recommendation can have a positive impact on both attracting the user's further participation and improving the crowdsensing system's effectiveness. However, such a personalized task recommender system is missing in the current crowdsensing literature. Jin *et al.* [7] and Wang *et al.* [14] studied the quality-aware incentive mechanism design without addressing the need of personalized

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303), in part by China NSF grant 61672348, 61672353, and 61472252, in part by Shanghai Science and Technology fund 15220721300, and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

F. Wu is the corresponding author.

task recommendation. Karaliopoulos *et al.* [8] proposed to assign the tasks to the users based on the profile of each user's probability of accepting a task, but did not consider the users' reliability information.

Central to the personalized task recommender system is a careful characterization on each user's preferences and reliability towards different tasks. However, it is not a trivial task, due to the unique nature of the crowdsensing scenarios. One of the challenges is finding a good way to model the users' preferences over different tasks. In some traditional recommender systems, customers' preferences can be readily obtained from their previous ratings [12]. However, the users in mobile crowdsensing do not typically provide explicit ratings on their preferences, *s.t.*, we have to infer the users' preferences from their implicit feedback, including the task browsing history, task selection record, and so on.

The most challenging part is estimating the users' reliability levels. In particular, we have to learn the users' reliability information for different tasks based on their submitted sensing data, if any, so as to build each user a profile characterizing the reliability levels of the users' data for performing the tasks. Although *truth discovery* algorithms [15] can be adopted to jointly estimate the users' data quality and the underlying truths, they cannot fully address the need of user reliability profiling in the context of task recommendation. We note that truth discovery algorithms usually generate a single reliability parameter for each user representing the overall trustworthiness level of the user. However, to conduct personalized task recommendation, the heterogeneity of a user's reliability in different tasks has to be reflected, and thus a more fine-grained reliability profiling of the users should be considered. A possible alternative is to independently generate each user a reliability parameter for each task by applying truth discovery algorithms to the data of each sensing task. Unfortunately, this approach may suffer from scalability issue, and what's worse, a user's reliability for a task cannot be estimated by truth discovery algorithms, if the user did not contribute data to that task. This could often be a problem in real crowdsensing scenarios, especially when the users' data are sparse, *i.e.*, each user only contributes data to only a small number of the tasks. Besides, without the prior knowledge of truth and reliability measures, typical truth discovery algorithms are likely to fail, when the majority of data are inaccurate [16].

In this work, we jointly consider the problems of user profiling and personalized task matching in mobile crowdsensing, and propose a personalized task recommender system framework, which recommends tasks to users based on both the users' preferences and reliability. We propose approaches to measure the users' preferences and reliability, respectively. First, in profiling the users' preferences, we introduce a hybrid preference metric that integrates the feedback against both the users' historical operations and the preference of their peers. Then, to tackle the more challenging part of profiling the users' reliability, we model the problem as a semi-unsupervised learning problem, and propose an efficient block coordinate descent algorithm to jointly estimate the users' reliability and the unknown ground truths. We surpass the existing truth discovery methods by not only taking the information of failed tasks into consideration but also using a small number of available truth data to facilitate the estimation accuracy. We further propose a matrix factorization method to address the missing entries in the users' reliability estimation. We conduct a real-world experiment and a large-scale crowdsensing simulation to evaluate the performance of our methods. The evaluation results show that our proposed methods can achieve superior performance over existing works and our benchmarks.

The main contributions of this work are listed as follows.

- First, we design a personalized task recommender system framework that matches tasks to users based on both the users' preferences and reliability levels of the tasks. We propose a method to profile each user's preferences over the tasks based on the user's implicit feedback.
- Second, we model the problem of user reliability profiling as a semi-supervised learning model, and propose an efficient algorithm to estimate the users' reliability and the unknown ground truths simultaneously. We also propose a matrix factorization method to estimate the users' reliability levels in their uninvolved tasks.
- Third, we conduct a real-world crowdsensing experiment and a large-scale simulation to evaluate the performance of our methods. Both the experiment and simulation results show that our proposed methods achieve dramatic performance improvements to our benchmarks.

The rest of the paper is organized as follows. We first present the system overview in Section II, and then introduce the problem formulations in Section III. In Section IV, we propose our reliability profiling algorithms. We evaluate our proposed methods and present the evaluation results in Section V. In Section VI, we review the related works. Finally, we conclude this paper in Section VII.

II. System Overview

In this section, we present an overview of our proposed personalized task recommender system.

A. System Model

Suppose there are N users and M sensing tasks in the system. The set of users and tasks are denoted by $\mathbb{N}$ and $\mathbb{S}$, respectively. We consider a *user-centric model*, where the users can browse the tasks in their smartphone applications and choose to participate in their interested tasks. If a user i wants to participate in a task j, she can click on some button to inform the platform her participation. After that, the user will use her smartphone to collect and then submit sensing data to the platform. Let $x_{i,j}$ denote the data submitted by the user i to the task j. The ground truth of the task j is denoted by x_j^*, which is usually unavailable to the platform.

We tend to build a personalized task recommender system, where the tasks are recommended to the users based on a joint consideration of the users' preferences and reliability. Specifically, for each task j, suppose each user i's preference and reliability regarding the task is denoted by $p_{i,j}$ and $q_{i,j}$, respectively. We propose a recommendation score $Score(i,j)$ that takes both the user i's preference and reliability for the task j into account, *i.e.*, $Score(i,j) = f(p_{i,j}, q_{i,j})$, where the function $f()$ outputs the recommendation score based on the two input parameters. For simplicity, we use a linear combination of the two parameters, *i.e.*,

$$Score(i,j) = \gamma\, p_{i,j} + (1-\gamma) q_{i,j}, \tag{1}$$

where γ is a hyper parameter. Other instances of the function f are possible, and the platform can determine the specific instance of the function according to its actual needs. We note that central to the system model is the users' preference and reliability measures. To that end, we need to carefully examine the historical data of the crowdsensing system, in order to acquire profiles of the users' preferences and reliability.

B. User Preference Profiling

To characterize the users' preferences on the tasks, the users' feedback information is needed. However, due to the unavailability on the users' explicit feedback (*e.g.*, ratings, like or dislike), implicit feedback has to be exploited. Fortunately, the crowdsensing platform can have access to each user's performance records on the applications, including which tasks the user has browsed, selected, or successfully completed. This information can be used to infer the users' preferences from two different perspectives, *i.e.*, either against the user's historical performance (content-based characteristics), or against the preferences of other similar users (collaborative-based characteristics) [12].

1) Content-Based Characteristic: Each task has many attributes, including time, location, travel distance, payment, and so on. Along with the users' task selection choices (selected or not), this information can be regarded as training examples. By using classification methods, such as logistic regression or Bayesian classifier, we can build a classifier to infer each user probability of selecting each task [8]. We let $P(i,j)$ denote the probability of the user i selecting the task j.

2) Collaborative-Based Characteristic: In mobile crowdsensing, the platform usually does not have users' ratings on tasks. Thus, implicit feedback from the users has to be exploited to infer the users' preferences. We let U denote the users' task preference matrix, where the entry $u_{i,j}$ means the user i's preference over the task j. The value of each $u_{i,j}$ can be calculated by mapping the user's implicit feedback to a task preference value, *i.e.*,

$$u_{i,j} = \begin{cases} \text{N/A} & \text{if } i \text{ did not browse task } j, \\ 0.5 & \text{if } i \text{ browsed but not selected task } j, \\ 1 & \text{if } i \text{ browsed and selected task } j. \end{cases} \tag{2}$$

The matrix U could be sparse, where many entries remain unknown. In this case, state-of-the-art collaborative filtering methods can be adopted to predict these missing entries [17].

To combine the two separate characteristics, we define each user i's preference for each task j as a linear combination of the content-based characteristic and the collaborative-based characteristic, *i.e.*,

$$q_{i,c} = \eta P(i,j) + (1-\eta) u_{i,j}, \tag{3}$$

where $\eta \in [0,1]$ is a constant parameter.

We note that many previous recommendation systems have investigated the problem of exploiting customers' implicit feedback in other application contexts (*e.g.*, [18], [19]), the intuitions of them can be further incorporated to improve our model of the users' preferences.

C. User Reliability Profiling

In the rest of the paper, we tend to put our most efforts on user reliability profiling, which is the most challenging part of the system. Given the set of collected sensing data, our objective is to jointly estimate the users' heterogeneous reliability levels for different tasks and the unknown ground truth values. An intuitive approach is to treat each task j independently and generate each user i a reliability measure $q_{i,j}$ for each task j. However, estimating each user's reliability based only on her data to a single task may be susceptible to noise, and thus cannot accurately reflect the user's reliability level. Besides, due to the large number of tasks, calculating a reliability parameter per user per task may not be efficient.

To tackle this problem, we tend to take the similarities among tasks into consideration by classifying the tasks into different categories, where the tasks within the same category focus on a similar sensing target. For example, some category only focuses on noise monitoring tasks, and some only focuses on traffic congestion monitoring. The classification of the tasks is common in current crowdsensing applications, *e.g.*, Waze [9]. It can be done by the platform's direct designation in the task release phase, or by applying text classification techniques [20] to automatically analyze the descriptions of the tasks. Specifically, we categorize the M tasks into C categories ($C \ll M$). For each category $c \in \{1,\dots,C\}$, the set of the tasks belong to the category is denoted by $\mathbb{S}_c$ ($\mathbb{S}_c \subseteq \mathbb{S}$). We assume that each task $j \in \mathbb{S}$ can only belong to one category, thus the sets $\mathbb{S}_1,\dots,\mathbb{S}_C$ are mutually disjoint. For each task category c, let $q_{i,c}$ denote each user i's reliability of the task category. Now, the user reliability profiling problem becomes to infer each user i's reliability $q_{i,c}$ in each category.

We note that different tasks may have different data types. For example, a task of weather report usually requires categorical data (*e.g.*, sunny, rainy, or cloudy), while a noise monitoring task may require continuous numerical data (*i.e.*, the noise levels of the users' surrounding environment). Thus, the reliability profiling algorithm needs to be carefully designed to handle both categorical and continuous data types.

III. Problem Formulation and Our Contributions

In this section, we present the problem formulation of user reliability profiling. We first present a preliminary version of our problem model, and then propose two enhancements. One enhancement is to incorporate the information of failed tasks, and the other is to integrate a small portion of truth data to improve the estimation accuracy.

A. Preliminary Problem Formulation

We assume that the tasks in different categories are independent, *s.t.*, we can estimate the users' reliability for each category separately. Let $\mathbb{N}_c$ denote the set of users who contributed data to tasks in category c. To estimate users' reliability, for each category c, we aim to solve the following optimization problem.

$$\min_{\{q_{i,c}\},\{\hat{x}_j^*\}} \sum_{i\in\mathbb{N}_c}\sum_{j\in\mathbb{S}_c} y_{i,j}\, q_{i,c}\, L(x_{i,j},\hat{x}_j^*), \quad \text{s.t. } \delta(\{q_{i,c}\})=1 \tag{4}$$

where $y_{i,j}$ indicates if the user i has contributed data to the task j, $\hat{x}_j^*$ is our estimation for the task j's ground truth, and $\delta()$ is a regularization function. Following the convention of truth discovery literature [21], we adopt the exponential regularization function, *i.e.*, $\delta(\{q_{i,c}\}) = \sum_{i\in\mathbb{N}_c}\exp(-q_{i,c})$. The loss function $L()$ measures the distance between a user's data and the estimated truth. For continuous data, $L()$ can be defined as the squared distance, *i.e.*, $L(x,\hat{x}^*) = (x-\hat{x}^*)^2$, while for categorial data, $L()$ can be defined as the $0-1$ distance, *i.e.*, $L(x,\hat{x}^*)=0$ if $x=\hat{x}^*$, and 1 otherwise. An intuitive interpretation of the problem formulation is that the ground truth should be close to the data contributed by reliable users, and the users whose data are close to the ground truth should be the reliable ones.

B. Contribution 1: Incorporating Information of Failed Tasks

We observe that in practice, the users may select certain tasks, but did not successfully complete them (*e.g.*, decide to terminate the sensing procedure half way). This phenomenon, referred as *failed tasks*, is likely to reflect the users' unreliability in performing certain tasks. In this part, we improve the above problem formalization by taking this issue into account.

We first introduce some notations. Among the set of tasks in category c, we let $\mathbb{S}_{i,c}$ denote the set of tasks the user i selected, and $\mathbb{D}_{i,c}$ the set of tasks the user i has successfully completed, where $\mathbb{D}_{i,c}\subseteq\mathbb{S}_{i,c}\subseteq\mathbb{S}_c$. For each category c, we calculate each user i's task completion ratio $r_{i,c}$, which is defined as the number of tasks the user i has finished over the number of tasks the user i has selected, *i.e.*, $r_{i,c}=\frac{|\mathbb{D}_{i,c}|}{|\mathbb{S}_{i,c}|}$. We revise the original formulation by multiplying a penalty term to $q_{i,c}$. The revised problem is presented as follows.

$$\min_{\{q_{i,c}\},\{\hat{x}_j^*\}} \sum_{i\in\mathbb{N}_c}\sum_{j\in\mathbb{S}_c} y_{i,j}\, q_{i,c}\, g(r_{i,c})\, L(x_{i,j},\hat{x}_j^*), \quad \text{s.t. } \sum_{i\in\mathbb{N}_c}\exp(-q_{i,c}\, g(r_{i,c}))=1, \tag{5}$$

where $g(x)=1-\log(x)$ is a function mapping each user's completion ratio to a penalty. We can see that the users who have failed tasks will receive a completion ratio less than 1, and thus their reliability outputs should be less than the ones estimated by the previous method shown in Equation 4. An extreme case is that some user i may select multiple tasks but completed zero (*i.e.*, $\mathbb{S}_{i,c}>0$ and $\mathbb{D}_{i,c}=0$). In this case, the system cannot generate a reliability estimation for the user. We will handle this problem in Section IV-B.

C. Contribution 2: Incorporating Available Ground Truths

The above formulation extends the basic truth discovery problem, which is built upon an underlying assumption that the majority of data are reliable. Unfortunately, it may suffer from a reliability initialization problem, *i.e.*, when most of the data are unreliable, the above estimation procedure may have bad performance [16]. To tackle this issue, we propose a *semi-supervised* learning framework, which incorporates a small number of ground truths to improve the estimation accuracy. To this end, the platform may intentionally add a few tasks with known ground truths into the task corpus to collect additional information on the users' reliability, whereas the users have no idea which tasks are inserted by the platform. The platform may also sample a few tasks, and employ some trusted workers to obtain their ground truths.

We let $\mathbb{S}$ denote the set of tasks with unknown ground truths, and $\mathbb{O}$ denote the set of tasks that are intentionally inserted by the platform with known truth information. For each category c of tasks, we let $\mathbb{S}_c$ and $\mathbb{O}_c$ denote the set of the tasks without and with prior ground truths respectively.

Having the ground truths of some tasks in hand, we propose to leverage those information to further enhance our estimation accuracy. To distinguish the notations, we let $\hat{x}_j^*$ denote the estimation of the ground truth ($j\in\mathbb{S}$), and x_o^* denote the known truth ($o\in\mathbb{O}$). Then, for each category c, the modified learning optimization problem is given by

$$\begin{aligned}\min_{\{q_{i,c}\},\{\hat{x}_j^*\}} \sum_{i\in\mathbb{N}_c} q_{i,c}\, g(r_{i,c})\Big(&\sum_{j\in\mathbb{S}_c} y_{i,j}\, L(x_{i,j},\hat{x}_j^*) \\ &+\alpha\sum_{o\in\mathbb{O}_c} y_{i,o}\, L(x_{i,o},x_o^*)\Big), \\ \text{s.t. } \sum_{i\in\mathbb{N}_c}\exp(-q_{i,c}\, g(r_{i,c}))=1,&\end{aligned} \tag{6}$$

where α is a hyper parameter controlling the relative weight of the second loss terms. We can see that the second loss term $\sum_{o\in\mathbb{O}_c} y_{i,o}\, L(x_{i,o},x_o^*)$ is constant for each user i in each task category c. We let $\epsilon_{i,c}$ denote the term $\sum_{o\in\mathbb{O}_c} y_{i,o}\, L(x_{i,o},x_o^*)$, and the problem presentation can be simplified as follows.

$$\min_{\{q_{i,c}\},\{\hat{x}_j^*\}} \sum_{i\in\mathbb{N}_c} q_{i,c}\, g(r_{i,c})\Big(\sum_{j\in\mathbb{S}_c} y_{i,j}\, L(x_{i,j},\hat{x}_j^*)+\alpha\epsilon_{i,c}\Big) \quad \text{s.t. } \sum_{i\in\mathbb{N}_c}\exp(-q_{i,c}\, g(r_{i,c}))=1. \tag{7}$$

IV. User Reliability Profiling Algorithm

In this section, we first propose a block coordinate descent algorithm to solve the user reliability profiling problem formulated above. Then, we further propose a matrix factorization method to estimate each user's reliability for the task categories that lack the user's historical performance.

A. Estimating Users' Reliability for Involved Categories

In our problem formulated in Equation 7, two sets of variables need to be estimated. We propose a block coordinate descent algorithm to solve it. The core idea of the algorithm is to fix one set of variables to solve the other, and repeat this process until convergence. Since the estimation process for each category can be done independently, parallel computing can be adopted to speed up the entire calculation process. For each task category c, we perform the following three steps.

*0) **Parameter Initialization**:* We first initialize the users' reliability $\{q_{i,c}\}$. Since a random or uniform initialization may result in poor estimation performance, which is especially true when most data are inaccurate, we propose to enhance the initialization stage by incorporation available ground truths. For each category c, let $\mathbb{N}_c^o$ denote the set of users who contributed data to tasks in $\mathbb{O}_c$. For the users in $\mathbb{N}_c^o$, their reliability can be initialized by solving the following problem.

$$\begin{aligned} \underset{\{q_{i,c}\}, i \in \mathbb{N}_c^o}{\text{argmin}} & \sum_{i \in \mathbb{N}_c^o} \sum_{o \in \mathbb{O}_c} y_{i,o} \, q_{i,c} \, g(r_{i,c}) \, L(x_{i,o}, x_o^*), \\ \text{s.t.} & \sum_{i \in \mathbb{N}_c^o} \exp(-q_{i,c} \, g(r_{i,c})) = \frac{|\mathbb{N}_c^o|}{|\mathbb{N}_c|}. \end{aligned} \tag{8}$$

As for the remaining users in $\mathbb{N}_c \setminus \mathbb{N}_c^o$, their reliability parameters are uniformly initialized such that

$$\sum_{i \in \mathbb{N}_c \setminus \mathbb{N}_c^o} \exp(-q_{i,c} \, g(r_{i,c})) = 1 - \frac{|\mathbb{N}_c^o|}{|\mathbb{N}_c|}. \tag{9}$$

Solving Equation 8 and Equation 9, we have the initialization of the users' reliability parameters, *i.e.*,

$$q_{i,c} = \begin{cases} \dfrac{\log\left(\frac{|\mathbb{N}_c| \sum_{i \in \mathbb{N}_c^o} \sum_{o \in \mathbb{O}_c} y_{i,o} L(x_{i,o}, x_o^*)}{|\mathbb{N}_c^o| \sum_{o \in \mathbb{O}_c} y_{i,o} L(x_{i,o}, x_o^*)}\right)}{g(r_{i,c})} & \text{if } i \in \mathbb{N}_c^o, \\ \dfrac{\log(|\mathbb{N}_c|)}{g(r_{i,c})} & \text{if } i \in \mathbb{N}_c \setminus \mathbb{N}_c^o. \end{cases} \tag{10}$$

Due to limitation of space, we put the details of solving the initialization problem into our technical report [22].

*1) **Truth Update**:* After obtaining an initial estimation of the users' reliability, we can update the estimation of truths by treating the estimated reliability parameters $\{q_{i,c}\}$ as fixed values. Then, the truth of each task $j \in \mathbb{S}_c$ can be updated using the following rule.

$$\{\hat{x}_j^*\} \leftarrow \underset{\{\hat{x}_j^*\}, j \in \mathbb{S}_c}{\text{argmin}} \sum_{i \in \mathbb{N}_c} q_{i,c} \, g(r_{i,c}) \Big(\sum_{j \in \mathbb{S}_c} y_{i,j} \, L(x_{i,j}, \hat{x}_j^*) + \alpha \epsilon_{i,c} \Big) \tag{11}$$

Theorem 1. *Given the users' reliability parameters, the optimization problem in Equation 11 can be optimally solved. For continuous data type, the optimal solution is given by*

$$\hat{x}_j^* = \frac{\sum_{i \in \mathbb{N}_c} q_{i,c} \, y_{i,j} \, x_{i,j} \, g(r_{i,c})}{\sum_{i \in \mathbb{N}_c} q_{i,c} \, y_{i,j} \, g(r_{i,c})}. \tag{12}$$

As for categorial data type, the solution is

$$\hat{x}_j^* = \underset{x'_j \in \{x_{i,j}\}}{\mathit{argmax}} \sum_{i \in \mathbb{N}_c} q_{i,c} \, y_{i,j} \, g(r_{i,c}) \, \mathbf{1}(x_{i,j}, x'_j), \tag{13}$$

where $\mathbf{1}(x, y) = 1$ if $x = y$, and 0 otherwise.

Proof. (*Sketch*) For either data type, we take partial derivative of the objective function with respect to x_j^* and set it to zero. Solving the equation, we can get the solution. Please refer to our technical report [22] for details. □

*2) **Reliability Estimation**:* After updating the estimation of the ground truth, we now fix the values of $\{\hat{x}_j^*\}$, and calculate the users' data qualities $\{q_{i,c}\}$ by solving the following optimization function. Intuitively, the users whose data are close to the ground truth estimations will have high reliability estimations, and vice versa.

Algorithm 1: User Reliability Estimation for Category c

Input: Tasks $\mathbb{S}_c$ and $\mathbb{O}_c$, users $\mathbb{N}_c$, and data $\{x_{i,j}\}$
Output: Reliability $\{q_{i,c}\}$, and truth estimation $\{\hat{x}_j^*\}$

- **if** $i \in \mathbb{N}_c$ **then**
  - **if** $i \in \mathbb{N}_c^o$ **then**
    - $q_{i,c} \leftarrow \frac{1}{g(r_{i,c})} \log\left(\frac{|\mathbb{N}_c| \sum_{i \in \mathbb{N}_c^o} \sum_{o \in \mathbb{O}_c} y_{i,o} L(x_{i,o}, x_o^*)}{|\mathbb{N}_c^o| \sum_{o \in \mathbb{O}_c} y_{i,o} L(x_{i,o}, x_o^*)}\right)$;
  - **else** $q_{i,c} \leftarrow \frac{\log(|\mathbb{N}_c|)}{g(r_{i,c})}$;
- **else** $q_{i,c} \leftarrow$ N/A;
- **while** *not converged* **do**
  - **foreach** *task* $j \in \mathbb{S}_c$ **do**
    - **if** *the task* j *is of continuous data type* **then**
      - $\hat{x}_j^* \leftarrow \frac{\sum_{i \in \mathbb{N}_c} q_{i,c} \, y_{i,j} \, x_{i,j} \, g(r_{i,c})}{\sum_{i \in \mathbb{N}_c} q_{i,c} \, y_{i,j} \, g(r_{i,c})}$;
    - **if** *the task* j *is of categorical data type* **then**
      - $\hat{x}_j^* \leftarrow \underset{x'_j \in \{x_{i,j}\}}{\text{argmax}} \sum_{i \in \mathbb{N}_c} q_{i,c} \, y_{i,j} \, g(r_{i,c}) \mathbf{1}(x_{i,j}, x'_j)$;
  - **foreach** *user* $i \in \mathbb{N}_c$ **do**
    - $q_{i,c} \leftarrow \frac{1}{g(r_{i,c})} \log\left(\frac{\sum_{i \in \mathbb{N}_c} \left(\sum_{j \in \mathbb{S}_c} y_{i,j} L(x_{i,j}, \hat{x}_j^*) + \alpha \epsilon_{i,c}\right)}{\sum_{j \in \mathbb{S}_c} y_{i,j} L(x_{i,j}, \hat{x}_j^*) + \alpha \epsilon_{i,c}}\right)$;
- **return** $\{q_{i,c}\}$ *and* $\{\hat{x}_j^*\}$

$$\begin{aligned} \{q_{i,c}\} \leftarrow \underset{\{q_{i,c}\}}{\text{argmin}} & \sum_{i \in \mathbb{N}_c} q_{i,c} \, g(r_{i,c}) \Big(\sum_{j \in \mathbb{S}_c} y_{i,j} \, L(x_{i,j}, \hat{x}_j^*) + \alpha \epsilon_{i,c} \Big) \\ \text{s.t.} & \sum_{i \in \mathbb{N}_c} \exp(-q_{i,c} \, g(r_{i,c})) = 1. \end{aligned} \tag{14}$$

Theorem 2. *Given fixed truth estimation $\{\hat{x}_j^*\}$, the problem in Equation 14 can be optimally solved. The optimal value of each $q_{i,c}, i \in \mathbb{N}_c$ is given by*

$$q_{i,c} = \frac{1}{g(r_{i,c})} log\Big(\frac{\sum_{i \in \mathbb{N}_c} \left(\sum_{j \in \mathbb{S}_c} y_{i,j} L(x_{i,j}, \hat{x}_j^*) + \alpha \epsilon_{i,c} \right)}{\sum_{j \in \mathbb{S}_c} y_{i,j} L(x_{i,j}, \hat{x}_j^*) + \alpha \epsilon_{i,c}} \Big). \tag{15}$$

Proof. (*Sketch*) We can see that the problem is convex. Therefore, we can apply the Lagrangian multiplier method to solve it. Due to limitation of space, we leave the details into our technical report [22]. □

The pseudo-code of the algorithm is presented in Algorithm 1. We first initialize the users' reliability parameters, and then keep iterating the steps of truth update and reliability estimation until convergence. Due to the convexity of our problem and the ability to achieve the optimal solution for each step (Theorem 1 and Theorem 2), our algorithm is guaranteed to converge to some local optimum, according to the proposition of the block coordinate descent [23]. Further improvements can be made to find a 2-approximation of the global optimum within nearly linear time [24].

B. Estimating Missing Entries: A Latent Factor Model

So far, we have obtained each user's reliability information over the task categories that she has contributed data to.

However, we observe that if a user i did not contribute data to some category c (*i.e.*, $i \notin \mathbb{N}_c$), then Algorithm 1 is not able to estimate the user i's reliability over c. In this part, we propose a matrix factorization method to address this problem.

We use Q to denote the users' reliability matrix, where each entry $q_{i,c}$ is the user i's reliability for task category c. We map both users and task categories to a joint latent factor space of dimensionality k. Specifically, we assume that each user i is associated with a vector $\boldsymbol{w}_i \in \mathbb{R}^k$, and each category is associated with $\boldsymbol{\theta}_c \in \mathbb{R}^k$. The vector $\boldsymbol{w}_i = [w_{i,1}, w_{i,2}, \ldots, w_{i,k}]^T$ can be interpreted as the user i's capabilities in k different dimensions, and the vector $\boldsymbol{\theta}_c = [\theta_{c,1}, \theta_{c,2}, \ldots, \theta_{c,k}]^T$ can be seen as the weight of each capability needed by the category c. Then, each user i's reliability for each category c can be calculated as $q_{i,c} = \boldsymbol{w}_i^T \boldsymbol{\theta}_c$.

To estimate the missing entries in matrix Q, we tend to calculate each user i's latent vector $\boldsymbol{w}_i$ and each category's latent vector $\boldsymbol{\theta}_c$. Let $\mathcal{W}$ and Θ denote the sets of users' and categories' latent vectors, respectively. Then, the objective function can be formalized as follows.

$$\min_{\mathcal{W},\Theta} \frac{1}{2} \sum_{c=1}^{C} \sum_{i=1}^{N} z_{i,c} (q_{i,c} - \boldsymbol{w}_i^T \boldsymbol{\theta}_c)^2 \tag{16}$$

where $z_{i,c}$ indicates if user i has contributed data to category c (1 means yes, and 0 otherwise). To prevent over-fitting, we add regularization terms in Equation 16.

$$\min_{\mathcal{W},\Theta} \frac{1}{2} \sum_{c=1}^{C} \sum_{i=1}^{N} a_{i,c} (q_{i,c} - \boldsymbol{w}_i^T \boldsymbol{\theta}_c)^2 + \frac{\lambda_1}{2} \sum_{i=1}^{N} \|\boldsymbol{w}_i\|^2 + \frac{\lambda_2}{2} \sum_{c=1}^{C} \|\boldsymbol{\theta}_c\|^2, \tag{17}$$

where $\|\boldsymbol{w}_i\|^2 = \sum_{t=1}^{k} w_{i,t}^2$ and $\|\boldsymbol{\theta}_c\|^2 = \sum_{t=1}^{k} \theta_{c,t}^2$. λ_1 and λ_2 are parameters controlling the weights of regularization terms.

We propose to use a simple gradient descent method to solve the above problem. The pseudo-code is presented in Algorithm 2. We first initialize $\{w_{i,t}\}$ and $\{\theta_{c,t}\}$ to small random values. After that, we apply gradient descent algorithm, *i.e.*, for every i and t, we update $\{w_{i,t}\}$ and $\{\theta_{c,t}\}$ using the following rules

$$w_{i,t} \leftarrow w_{i,t} - \beta\Big(\sum_{c=1}^{C} z_{i,c}(q_{i,c} - \boldsymbol{w}_i^T \boldsymbol{\theta}_c) + \lambda_1 w_{i,t}\Big), \tag{18}$$

$$\theta_{c,t} \leftarrow \theta_{c,t} - \beta\Big(\sum_{i=1}^{N} z_{i,c}(q_{i,c} - \boldsymbol{w}_i^T \boldsymbol{\theta}_c) + \lambda_2 \theta_{c,t}\Big), \tag{19}$$

where β is the learning rate. Finally, we can predict a user i's reliability for a task category c even if the user i did not provide any data to c, *i.e.*, for $i \notin \mathbb{N}_c$, $q_{i,c} \leftarrow \boldsymbol{w}_i^T \boldsymbol{\theta}_c$.

V. Evaluation

In this section, we implement and evaluate the performance of our proposed methods. We first conduct a real-world crowdsensing experiment, and then simulate a large-scale scenario to further examine the performance of our methods.

A. Experiment Setup

We recruit 10 users (8 males and 2 females) to participate in our experiment. In the experiment, we manually create 123 sensing tasks for 9 different categories. The tasks within the same category focus on the same sensing target (such as noise, traffic, or weather), but with different attributes, including time, locations, and payments. Each task category has a data type requirement. For instance, noise monitoring requires continuous data type, while weather monitoring requires categorical data type. The entire task corpus is shown to the users through the browsers on the users' smartphones. Each user can browse through these tasks, and choose their interested tasks to work on. The ground truth of each task is monitored by the authors themselves, and unavailable to the users. We collect the users' sensing data, as well as their operation records, including each user's task browsing history, task selection history, and task completion history.

Algorithm 2: Unknown Reliability Estimation

Input: Users reliability matrix Q
Output: Unknown reliability parameters $\{q_{i,c} | z_{i,c} = 0\}$

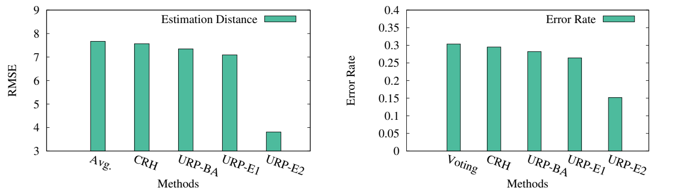
```
Initialize {w_i} and {θ_c} to small random values;
while not converged do
    foreach i=1,...,N, c=1,...,C do
        w_{i,t} ← w_{i,t} − β( Σ_{c=1}^{C} z_{i,c}(q_{i,c} − w_i^T θ_c) + λ_1 w_{i,t} ),
        θ_{c,t} ← θ_{c,t} − β( Σ_{i=1}^{N} z_{i,c}(q_{i,c} − w_i^T θ_c) + λ_2 θ_{c,t} );
foreach q_{i,c} = N/A do
    q_{i,c} ← w_i^T · θ_c
return {q_{i,c} | z_{i,c} = 0}
```

According to our collected data, each user contributes data to about 60% of the tasks in average. The parameter α used in our semi-supervised learning model is set to 1. And for each task category, we use the ground truths of 10% of the tasks. The parameters k, $\lambda 1$ and λ_2 used in our matrix factorization method are set to 3, 5 and 5, respectively.

B. Experiment Results on User Reliability Profiling

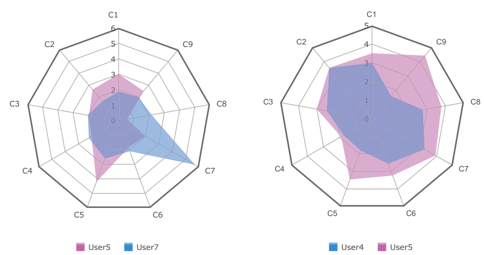
In the experiment, we evaluate the performance of our proposed user profiling algorithm. To differentiate the notations, we use "URP-BA" to denote the basic version shown in III-A, and "URP-E1" and "URP-E2" to denote the first enhancement and the second enhancement, respectively. We compare our algorithms with two benchmarks. One is a heuristic method that treats each user's data equally, *i.e.*, simple average ("Avg.") for continuous data and majority voting ("Voting") for categorical data. The other benchmark is a general truth discovery framework, called "CRH" [21], which uses a single parameter to model each user's reliability level. We adopt the following two metrics to measure the performance of the algorithms.

- RMSE: For continuous data, we use Root Mean Square Error (RMSE) to measure the distance between the estimation result and the ground truth. Mathematically, the RMSE is defined as $\sqrt{\sum_{j \in \mathbb{S}} (x_j^* - \hat{x}_j^*)^2 / |M|}$.
- Error Rate: For categorical data, we use Error Rate to quantify the performance of an algorithm. The Error Rate of an algorithm is defined as the percentage of the tasks to which the algorithm's estimations are different from the ground truth, *i.e.*, $1 - \frac{\sum_{j \in \mathbb{S}} \mathbf{1}(x_j^*, \hat{x}_j^*)}{M}$.



(a) Continuous Data (b) Categorical Data

Fig. 1. Performance Comparison on Estimation Accuracay



(a) Reliability (b) Preference

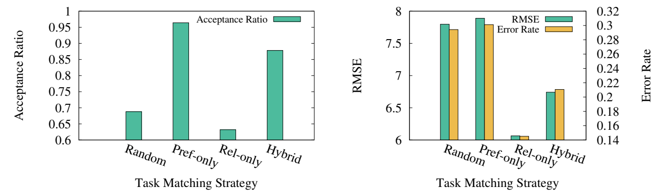
Fig. 2. User Profiling

Fig. 1 presents the performance comparison between our algorithms and the benchmarks. We can see that for either data type, the truth discovery-based algorithms can achieve higher estimation accuracy than the simple average or majority voting, indicating the effectiveness of truth discovery algorithms. However, the performance of Avg./Voting, CRH, URP-BA, and URP-E1 tends to be similar. The main reason is that under the crowdsensing scenarios, these usually exist many tasks to which the majority of the users' data are inaccurate, thus the traditional unsupervised learning models may have trouble identifying the users' true reliability levels. In this case, as we can see that URP-E2 has superior performance to the other four algorithms, incorporating even a small number of ground truths can dramatically improve the estimation accuracy.

C. Experiment Results on Personalized Task Matching

Besides profiling the users' reliability, we also profile each user's preference towards each task using the methods proposed in Section II-B. In Fig. 2(a) and Fig. 2(b), we present the reliability profiles and preference profiles of two representative users respectively, where the user's preference towards a task category is calculated as the user's average preference score of the tasks in the category. We normalize the users' preferences to [0,5] for better graphical presentation.

To evaluate the performance of our personalized task recommender system, we provide each user a list of 20 recommended tasks, and ask each user to choose their interested tasks. Recall that our personalized task recommender system recommends tasks to the users based on both the users' reliability and preference. Specifically, for each user and task pair (i, j), we calculate a recommendation score $Score(i, j) = \gamma p_{i,j} + (1 - \gamma) q_{i,j}$. Suppose task j belongs to category c, then we set $p_{i,j}$ to $p_{i,c}$. We use $\gamma = 0.4$ and $\eta = 0.5$ in our experiment. After that, our system recommends each user 20 tasks with the highest recommendation scores. Three benchmarks are adopted, including random recommendation, preference-only recommendation, and reliability-only recommendation. Random task recommendation strategy provides each user a list of 20 randomly chosen tasks, while the preference- or reliability-only recommendation strategies provide each user 20 tasks with highest preference or reliability scores, respectively.



(a) Acceptance Ratio (b) Estimation Accuracy

Fig. 3. Comparison on Different Task Matching Strategies

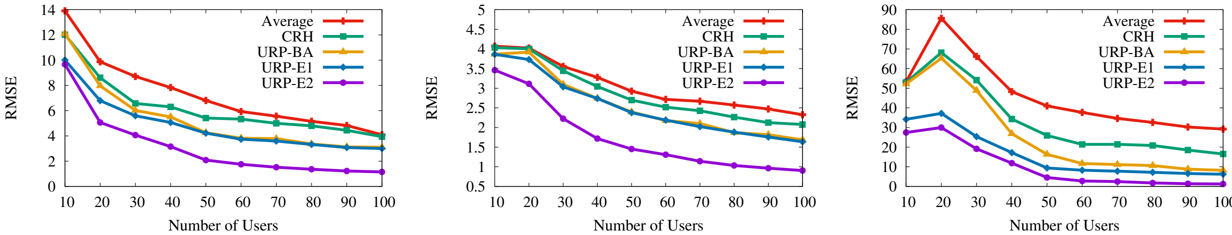
The performance of task matching strategies is measured on two different perspectives, *i.e.*, task acceptance ratio and estimation accuracy. The task acceptance ratio is defined as the percentage of the recommended tasks that the users have selected, and the estimation accuracy is measured using RMSE or Error Rate depending on the data types of the tasks. The performance comparison of different task matching strategies is presented in Fig. 3. We can see that the preference-only strategy has the highest task acceptance ratio, while the reliability-only strategy outputs the most accurate estimation results. That is because these two strategies match tasks to the users with the tendency of facilitating the match of one certain perspective. Comparing with other task matching strategies, we can see that our proposed hybrid recommendation strategy can achieve a good balance between the acceptance ratio and the estimation accuracy.

D. Evaluations on A Large-Scale Scenario

In this subsection, we examine the performance of our user profiling algorithm on a large-scale crowdsensing scenario.

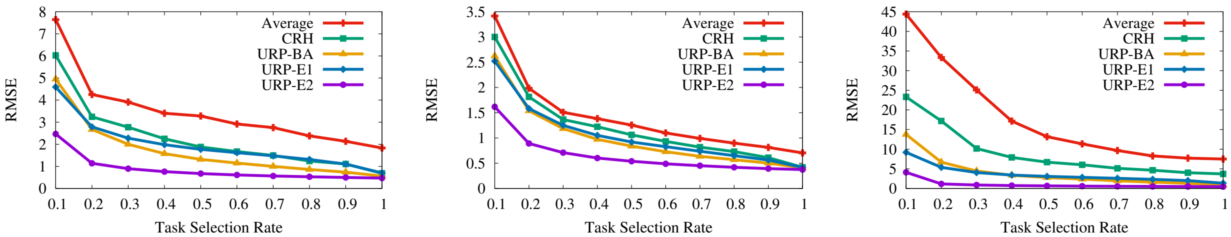
In our simulation, there are 100 users and 1000 tasks. These tasks are randomly distributed among 20 categories. Each user's task selection rate is set to 10%, *i.e.*, each user contributes data to each task with 10% probability. The ground truth of each task is randomly distributed within [30,100]. For each user i, if she contributes data to the task j of category c, then her data $x_{i,j}$ is generated based on a Gaussian distribution with the mean x_j^* and variance $\frac{2}{q_{i,c}}$, *i.e.*, $x_{i,j} \sim \mathcal{N}(x_j^*, \frac{2}{q_{i,c}})$. In URP-E2, we randomly choose 1% of tasks, and incorporate their ground truths in the user reliability profiling process. All the results are averaged over 1000 rounds.

We classify the users into three groups: reliable users, normal users, and unreliable users, where the users' reliability parameter in these three groups are assumed to follow $\mathcal{N}(0.75, 0.1)$, $\mathcal{N}(0.5, 0.1)$, and $\mathcal{N}(0.25, 0.1)$, respectively. We consider three different settings. In the first setting, the users are classified into the three groups randomly. In the second setting, each user has 60% probability of being classified into reliable users, 30% normal users, and 10% unreliable users, while in the third setting, each user has 10% being reliable, 30% being normal, and 60% being unreliable. We assume that for each user, if her reliability for certain task is below 0.2, then the user will have 50% probability of failing the task.



(a) Setting1 (b) Setting2 (c) Setting3

Fig. 4. Comparisons on estimation accuracy with varying number of users



(a) Setting1 (b) Setting2 (c) Setting3

Fig. 5. Comparisons on estimation accuracy with varying users' task selection rate

Fig. 4 presents the estimation accuracy of different algorithms with a varying number of the users. The number of users varies from 10 to 100 with the increment of 10. We can see that the simple average has the worst estimation accuracy, while URP-E2 achieves the lowest RMSE in all the three settings. In 4(c), we observe that the RMSE first grows as the number of users increases, and then decrease when the number of users is getting larger. This is because that when the number of users is small, slightly increasing the number of users, especially unreliable users, may bring extra errors to the estimation results. As the number of users increases, the platform can access to more information, and thus can reduce the estimation errors.

Fig. 5 shows the estimation accuracy of different algorithms with varying task selection rate. We increase the task selection rate from 0.1 to 1 with the increment of 0.1. It can be seen that our proposed user profiling algorithm achieves the lowest RMSE, indicating the effectiveness of our algorithm. Besides, we can observe that the RMSE decreases as the task selection rate increases. This is because that increasing the task selection rate usually means having more data, *s.t.*, the platform can identify the users' reliability levels more accurately. A similar phenomenon was also observed in [25].

We also examine the effect of the number of incorporated ground truths on the estimation accuracy. The results are shown in Fig. 6. We can see that having more truth can improve our estimation results. Besides, comparing the different settings, we can see that Setting 2 achieves the best estimation accuracy, since most users in Setting 2 are reliable.

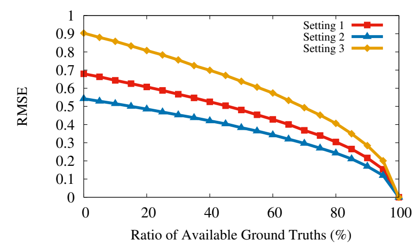


Fig. 6. The Effect of Available Truth on Estimation Accuracy

VI. Related Work

Many researchers have studied the user selection problem in mobile crowdsensing from the game-theoretic perspective. Yang *et al.* [2] proposed incentive mechanisms for both platform-centric model and user-centric model. Zhao *et al.* [3] considered the problem of budget feasible mechanism design for crowdsensing, and proposed mechanisms for both offline and online scenarios. He *et al.* [26] studied the optimal task allocation problem for location-dependent crowdsensing. Karaliopoulos *et al.* [8] adopted logistic regression techniques to estimate a user's probability of accepting a task, and tend to match tasks to users based on the information. However, none of these work considered the users' data quality or reliability in performing the sensing tasks. Although Jin *et al.* [7] and Han *et al.* [27] considered the problem of quality-aware task matching, they were based on the platform-centric model, and were unable to recommend personalized tasks for the users.

The problem of truth discovery has been widely studied to handle the situation where data collected from multiple sources tend to be conflicting and the ground truths are unknown [15]. Wang *et al.* [28] considered the problem of truth detection in social sensing based on EM algorithm. Wang *et al.* [29] proposed a truth discovery algorithm to handle streaming data. Ouyang *et al.* [30] proposed a truth discovery method to detect spatial events based on a graphical model. Su *et al.* [31] designed a generalized decision aggregation framework for distributed sensing scenarios. Wang *et al.* [32] studied the truth discovery problem in cyber-physical systems. Wang *et al.* [33] further exploited the problem of truth discovery for interdependent phenomena in social sensing. Meng *et al.* [34] exploited the spatial correlations to improve the estimation accuracy. CRH [21] is a general truth discovery framework that can handle both continuous and categorical data. Li *et al.* [25] considered truth discovery problem for long-tail data, and proposed a confidence-aware approach. Ma *et al.* [35] proposed a probabilistic method to tackle the scenarios where sources' reliability vary among different topics. Yang *et al.* [36] studied the problem of data quality estimation and quality-based payment determination. Peng *et al.* [37] propose an EM algorithm to quantity the users' data qualities in mobile crowdsensing. However, all of these works are based on unsupervised learning models, and thus may suffer from the reliability initialization problem when most data are inaccurate [16]. Yin and Tan *et al.* [38] proposed a semi-supervised learning model to identify true facts from false ones. However, their work tended to focus on the truth estimation part, but did not output the reliability levels of the data sources, thus cannot address the need of user reliability profiling.

VII. Conclusion

In this paper, we have studied the problem of personalized task matching in mobile crowdsensing. We have proposed a personalized task recommender framework that can recommend tasks to users based on a fine-grained characterization on both the users' preference and reliability. We have proposed methods to measure each user's preferences and reliability of different tasks, respectively. In particular, the proposed user reliability profiling algorithm originates from truth discovery problem, but surpasses existing truth discovery algorithms in two ways, *i.e.*, by exploiting the information of failed tasks and also by incorporating a small number of ground truths to improve the estimation accuracy. Further more, we proposed a matrix factorization method to address a critical limitation of the existing truth discovery algorithms in estimating the users' reliability for the uninvolved tasks. Both a real-world experiment and a large-scale simulation have been conducted to evaluate our proposed methods. The evaluation results have demonstrated the good performance of our methods.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[2] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *MobiCom*, 2012.

[3] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *INFOCOM*, 2014.

[4] Q. Zhao, Y. Zhu, H. Zhu, J. Cao, G. Xue, and B. Li, "Fair energy-efficient sensing task allocation in participatory sensing with smartphones," in *INFOCOM*, 2014.

[5] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *INFOCOM*, 2015.

[6] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun, "Incentive mechanism for proximity-based mobile crowd service systems," in *INFOCOM*, 2016.

[7] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *MobiHoc*, 2015.

[8] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias, "First learn then earn: optimizing mobile crowdsensing campaigns through data-driven user profiling," in *MobiHoc*, 2016.

[9] Waze. [Online]. Available: https://www.waze.com/

[10] Field agent. [Online]. Available: http://www.fieldagent.net/

[11] Gigwalk. [Online]. Available: http://www.gigwalk.com/

[12] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[13] K. L. Huang, S. S. Kanhere, and W. Hu, "Are you contributing trustworthy data?: the case for a reputation system in participatory sensing," in *MSWiM*, 2010.

[14] J. Wang, J. Tang, D. Yang, E. Wang, and G. Xue, "Quality-aware and fine-grained incentive mechanisms for mobile crowdsensing," in *ICDCS*, 2016.

[15] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[16] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han, "A survey on truth discovery," *Acm Sigkdd Explorations Newsletter*, vol. 17, no. 2, pp. 1–16, 2016.

[17] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, 2009.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*, 2009.

[19] C. H. Lin, E. Kamar, and E. Horvitz, "Signals in the silence: Models of implicit feedback in a recommendation system for crowdsourcing." in *AAAI*, 2014.

[20] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1289–1305, 2003.

[21] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *SIGMOD*, 2014.

[22] S. Yang, K. Han, Z. Zheng, S. Tang, and F. Wu. (2017, December) Technical report. [Online]. Available: https://www.dropbox.com/s/omr9mbmw2liipo6/report10.pdf?dl=0

[23] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[24] H. Ding, J. Gao, and J. Xu, "Finding global optimum for truth discovery: Entropy based geometric variance," in *SoCG 2016)*, 2016.

[25] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.

[26] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *INFOCOM*, 2014.

[27] K. Han, H. Huang, and J. Luo, "Posted pricing for robust crowdsensing," in *MobiHoc*, 2016.

[28] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *IPSN*, 2012.

[29] D. Wang, T. Abdelzaher, L. Kaplan, and C. C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *ICDCS*, 2013.

[30] R. W. Ouyang, M. Srivastava, A. Toniolo, and T. J. Norman, "Truth discovery in crowdsourced detection of spatial events," in *CIKM*, 2014.

[31] L. Su, Q. Li, S. Hu, S. Wang, J. Gao, H. Liu, T. F. Abdelzaher, J. Han, X. Liu, Y. Gao *et al.*, "Generalized decision aggregation in distributed sensing systems," in *RTSS*, 2014.

[32] S. Wang, D. Wang, L. Su, L. Kaplan, and T. F. Abdelzaher, "Towards cyber-physical systems in social spaces: The data reliability challenge," in *RTSS*, 2014.

[33] S. Wang, L. Su, S. Li, S. Hu, T. Amin, H. Wang, S. Yao, L. Kaplan, and T. Abdelzaher, "Scalable social sensing of interdependent phenomena," in *IPSN*, 2015.

[34] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *SenSys*, 2015.

[35] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han, "Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation," in *SIGKDD*, 2015.

[36] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.

[37] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *MobiHoc*, 2015.

[38] X. Yin and W. Tan, "Semi-supervised truth discovery," in *WWW*, 2011.