Optimal Task Recommendation for Mobile Crowdsourcing with Privacy Control

Yanmin Gong, *Student Member, IEEE,* Linbo Wei, *Member, IEEE,* Yuanxiong Guo, *Member, IEEE,* Chi Zhang, *Member, IEEE* and Yuguang Fang, *Fellow, IEEE*

***Abstract*—Mobile crowdsourcing (MC) is a transformative paradigm that engages a crowd of mobile users (i.e., *workers*) in the act of collecting, analyzing, and disseminating information or sharing their resources. To ensure quality of service, MC platforms tend to recommend MC tasks to workers based on their context information extracted from their interactions and smartphone sensors. This raises privacy concerns hard to address due to the constrained resources on mobile devices. In this paper, we identify fundamental trade-offs among three metrics–utility, privacy, and efficiency–in a MC system and propose a flexible optimization framework that can be adjusted to any desired trade-off point with joint efforts of MC platform and workers. Since the underlying optimization problems are NP-hard, we present efficient approximation algorithms to solve them. Since worker statistics are needed when tuning the optimization models, we use an efficient aggregation approach to collecting worker feedbacks while providing differential privacy guarantees. Both numerical evaluations and performance analysis are conducted to demonstrate the effectiveness and efficiency of the proposed framework.**

***Index Terms*—Privacy, Mobile crowdsourcing, task recommendation, differential privacy.**

I. Introduction

Mobile crowdsourcing (MC) is the combination of crowdsourcing and mobile technologies that leverages the advanced sensing, computing, and communication capabilities of mobile devices to provide crowdsourcing services. In MC, a crowd of mobile users are engaged to provide pervasive and cost-effective services of data collecting, processing, and computing. These mobile users have shifted from the traditional role of service consumers to the new role of service providers, and they usually collect a small fee (or other forms of reward) for providing services. The applications of mobile crowdsourcing have developed rapidly. Existing commercial MC applications include traffic monitoring (e.g., Waze [1]), ride sharing (e.g. Uber [2]), environmental monitoring (e.g., Stereopublic [3]), and wireless coverage mapping (e.g. OpenSignal [4]). Nonetheless, MC is still in its infancy, and there are many undergoing research exploring applications such as epidemics monitoring and prediction [5] and urban sensing [6]. Most of these applications are Internet of Things (IoT) systems, where a huge number of physical machines are connected over networks. MC can be a helpful technique to achieve the high-scale interconnectivity and to ensure the security, reliability, and cost-efficiency in these IoT systems.

In MC, a spatio-temporal task is outsourced to a group of mobile users (i.e., *workers*) who perform the task within a deadline, and only workers under certain contexts are qualified for the task. However, it is quite inefficient for workers to select tasks by themselves when there are a huge number of crowdsourcing tasks, especially on a mobile device due to its limited screen and keyboard. Hence, MC platforms must provide task recommendation services which proactively push a task to qualified workers. In current solutions, workers have to reveal their exact contexts to MC platforms in order to receive personalized task recommendation.

Depending on the application scenario, the context of a worker can be defined with multiple dimensions, including geographical (e.g., on a street), temporal (e.g., within hours), activity (e.g., moving speed), and profile (e.g., gender) [7]. These contexts contain private and sensitive information that may be used to uniquely identify an individual, reveal his/her health status, or track his/her daily routines. However, MC platforms are potentially untrustworthy in the sense that they may be operated by various organizations and companies and may also be compromised by malicious adversaries. Hence, allowing MC platforms to learn exact contexts may put workers' privacy at risk [8]. It is imperative to protect workers' privacy in order to enable large-scale deployments of mobile crowdsourcing applications.

An MC system has three components that may reveal private worker information: *offline statistics collection* to learn recommendation rules based on worker contexts and historical task completion performance, *online task selection* to select the most suitable tasks to a worker based on his current context, and *task completion* for a worker to accept and perform a task, and to return the result back. Each component exposes

Y. Gong and Y. Fang are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA.
E-mail: {ymgong@, fang@ece.}ufl.edu.

L. Wei and C. Zhang are with the Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, and the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China. E-mail: {lingbowei, chizhang}@ustc.edu.cn.

Y. Guo is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA.
E-mail: richard.guo@okstate.edu.

This work was partially supported by US National Science Foundation under grant CNS-1423165 and CNS-1409797, and the Natural Science Foundation of China under grant 61328208. The work of C. Zhang was also partially supported by the Natural Science Foundation of China under grant 61202140, by the Program for New Century Excellent Talents in University under grant NCET-13-0548, by the Innovation Foundation of the Chinese Academy of Sciences under grant CXJJ-14-S132, and by the Fundamental Research Funds for the Central Universities under grant WK2101020006.

A preliminary version of this work has been published in IEEE GLOBECOM'14.

worker contexts and raises privacy concerns in different ways. Privacy protection of the last component can be provided through anonymous routing or pseudonyms such as Tor and is not our focus in this paper. In this paper, we focus on the privacy issues in the first two components, and propose a framework for protecting privacy of worker contexts while enabling effective task recommendation in MC systems. The framework consists of two main components that may operate in parallel: *privacy-aware online task selection* which selects the best MC tasks for workers based on their current contexts, and *privacy-preserving offline statistics collection* which aggregates historical information about worker contexts and task completion activities needed for task selection while preserving privacy.

Privacy-Aware Online Task Selection. Current MC systems select tasks by collecting personal data at a server. Workers have to reveal their exact context information to the server in order to participate. To address the privacy concerns of such *server-only* recommendation, an alternative approach would be *worker-only*, where workers' mobile devices keep their own personal context information and perform recommendation. Indeed, it has been proposed for personalization in mobile advertising systems [9]. The problem with this approach is the huge computation and communication overhead for resource-constrained mobile devices. Thus, some recent papers propose hybrid solutions that jointly consider both sides to address privacy issues in mobile systems [10]–[13]. For example, in [10], the server returns a superset of the results and let end users to filter useful information by themselves. These solutions have a variety of optimization goals, which motivates us to consider the fundamental trade-offs in these mobile systems.

In this paper, we formulate the task selection from an MC server to a worker as an optimization problem that considers three metrics: (1) privacy that is related to the amount of a worker's context information shared with the MC server, (2) utility that represents the benefits of recommending the tasks, and (3) efficiency that measures the communication and computation overhead imposed on a worker's mobile device by recommending a certain number of tasks. We show in Section III that these three metrics cannot be optimized simultaneously. Note that the aforementioned solutions only present tradeoffs for certain instances: recommendation only at the server side provides the best efficiency and utility at the cost of privacy, while recommendation at the worker side provides privacy guarantee and utility at the cost of efficiency. In contrast, we propose an optimization model that can be adjusted to any desirable trade-off level. In the proposed optimization framework, a worker can decide how much information about his/her context to share with the MC server. Based on this limited information, the MC server selects and sends a set of tasks to the worker. The size of the task set is pre-defined by the worker considering the associated communication and computation overheads. After the worker receives the task set, he/she picks and completes the best task based on his private information. The most challenging part in the whole process is to select the task set sent by the MC server that maximizes the total expected utility of the MC server given the constraints on privacy and efficiency. There are also other types of tradeoffs we can consider, such as jointly optimizing utility and efficiency given a constraint on privacy. Since the priorities of privacy and efficiency can be arbitrarily selected by the worker, the framework is quite flexible and can be used in different MC systems.

Privacy-Preserving Offline Statistics Collection. Recommended tasks are chosen based on statistics including both historical performance of workers and the distribution of their contexts. These statistics can be collected offline and are used to calibrate the online task selection component. However, extracting these statistics often poses a privacy challenge: workers may be unwilling to reveal the required information such as their exact contexts and tasks that they have completed successfully. Therefore, we need to provide a privacy-preserving solution that can obtain these statistics from distributed worker data. Some previous works propose to address privacy in statistical queries by anonymizing data; however, there are possibilities that data owners may be de-anonymized with auxiliary information [14], [15]. Differential privacy adds noise in the querying results of statistical databases so that even with auxiliary information, one cannot infer the presence or absence of individuals. In this paper, we use a privacy-preserving statistics collection approach to reliably computing the required statistics from a dynamic set of workers who are potentially malicious. Our solution is based on a distributed statistics collection protocol provided in [16], which uses a semi-honest third party to add blind differentially private noise to distributed worker data.

The main contributions of this paper are as follows.

- We identify specific privacy challenges of task recommendation in MC systems, and then propose a framework that protects worker context privacy.
- We develop an optimization model for task selection that explores fundamental trade-offs among three design metrics–privacy, utility, and efficiency–in MC systems, and then present efficient approximation algorithms to solve it.
- We use an efficient statistics collection approach to preserving differential privacy in a distributed setting with tolerance of malicious or dynamic workers.
- We conduct both numerical evaluations and performance analysis to demonstrate the effectiveness and efficiency of our proposed framework.

The remainder of this paper is organized as follows. We first present our framework in Section II. Then we represent the task selection process as a constrained optimization problem in Section III. Section IV gives an approximation algorithm to solve the optimization problem. A privacy-preserving approach for statistics collection is presented in Section V. We discuss the experimental results and analyze the system overhead in Section VI and Section VII, respectively. Section VIII summarizes the related work. Finally, we conclude the paper in Section IX.

II. The Proposed Framework

In this section, we describe the basic system model for task recommendation in MC systems and design goals.

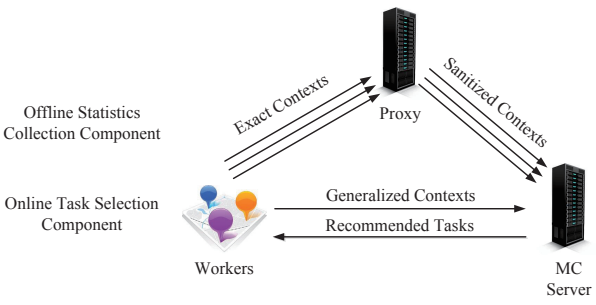


Fig. 1: Basic system model for task recommendation in MC.

A. System Model

Fig. 1 shows the basic model of the proposed framework consisting of the following two components:

- *Statistics Collection.* In this component, the server collects various statistics from workers periodically in the background. A semi-honest third party (to be elaborated later in Section V) is employed to protect the private context information of participating workers.
- *Task Selection.* In this component, based on the statistics collected in the statistics collection component and worker's current context, the server selects and delivers a set of tasks to the worker. Note that we allow workers to decide how much private information they are willing to share with the server. The server selects a set of tasks, where the set size is constrained by a bounded communication overhead, based on this limited information and sends them to the worker. The worker [1] then chooses the most relevant one to complete based on all his private information and returns the answer to task requesters.

Privacy Guarantees. Our framework can protect worker privacy in both online task selection and offline statistics collection. Note that task selection and statistics collection use private worker contexts in different ways, and therefore require different privacy-preserving techniques.

In *task selection*, a single worker's current context is used, and we ensure worker privacy through limited information disclosure as used in many mobile systems [12], [17], [18]. We allow the worker to share a generalized context with the server rather than his exact context. The generalization of worker context is done according to a predefined hierarchy. Quantifiable contexts such as location can be simply divided into different intervals based on their values. For instance, location information represented by the latitude and longitude with a total of 6 decimal digits can be generalized by keeping $6 - a$ decimal digits for level-a generalization. A worker can also choose different (i.e., adaptive) levels of generalization for different intervals of contexts with existing approaches [10].If a context information is not quantifiable, the generalization rule can be pre-defined. For example, user activity can be generalized based on a tree taxonomy as shown in Fig. 2. The pre-defined taxonomy are stored at the mobile device of

[1]For brevity, we use "he" to refer to the worker without meaning any distinctions about the worker's gender in the rest of the paper.

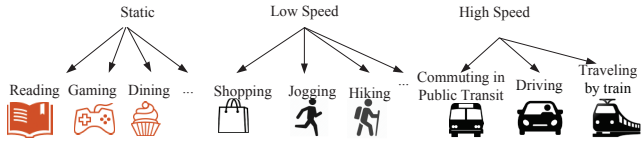


Fig. 2: Generalization of user activity.

workers, and a worker can reveal that he is moving at low speed rather than he is shopping. Interesting readers may refer to [19] for details about different generalization methods.

In *statistics collection*, historical context and task completion information from workers is used. Our framework allows workers to choose whether to participate in statistics collection, and protect the privacy of participating workers. We consider the privacy risk for participating workers from two aspects. We first guarantee that no other party, except the worker himself, would know his private information during statistics collection, which can be achieved through data encryption [20]–[22].

Moreover, we also consider privacy leakage that cannot be solved by data encryption. A potential privacy leakage is due to multiple runs of statistics collection when a worker does not participate in all runs, e.g., because he has reached home. Hence, we should protect every worker from an adversary (with arbitrary background knowledge) who tries to trace or de-anonymize a user between several runs of the statistics collection approach. To this end, we adopt the privacy notion of (ϵ, δ)-differential privacy [23], which ensures that the result of our approach does not significant change with the presence or absence of a single worker. Formally speaking,

Definition 1. A statistics collection algorithm $\mathcal{F}$ satisfies (ϵ, δ)-differential privacy if for any two datasets D_1 and D_2 which differ on at most one element, and $\forall O \subseteq \text{range}(\mathcal{F})$, the following inequality holds:

$$\Pr[\mathcal{F}(D_1) \in O] \leq \exp(\epsilon) \times \Pr[\mathcal{F}(D_2) \in O] + \delta. \quad (1)$$

In the definition, the parameter ϵ bounds the ratio of probability distributions of two datasets differing on at most one element, while δ permits us to relax the relative shift at events that are not likely to happen, bounding the probability of a privacy breach. In order to achieve (ϵ, δ)-differential privacy, the raw output of statistics computation algorithm $\mathcal{F}$ is sanitized by adding noise to it. The process of noise addition will be described in detail in Section V. If the outputs of statistics collection module achieve (ϵ, δ)-differential privacy, the fact whether a worker provides information or not to the recommendation server will not change the server's knowledge on him. Therefore, an adversary with arbitrary background knowledge cannot trace or de-anonymize a worker from multiple runs of the statistics collection approach.

B. Design Goals

For task selection, we aim to provide good privacy, utility, and efficiency.

- *Privacy*. Worker contexts are needed for task recommendation, which may be leveraged by the server to

uniquely identify an individual worker. To reduce the risk of being identified, the worker limits the information shared with the server. Instead of providing an exact context, the worker provides a generalized context which obfuscates privacy sensitive information such as location and activity.

- *Utility*. Utility is an abstract term which represents the value of a set of recommended tasks. It should be optimized during the task recommendation process. In this paper, the utility of the server is defined as the expected revenue (or commission) of the recommended tasks, while the utility of the worker is defined as the payment he would obtain by completing the recommended tasks. The utility for both stakeholders is related to the payment of the task that is selected and completed successfully by the worker.
- *Efficiency*. When a worker receives a set of recommended tasks, he tries to select the best task from the set. A larger set takes more time to select from, which contradicts the intention of recommendation. Thus the efficiency of task recommendation is directly related to the set size. The recommendation system should recommend a reasonable number of tasks at a time to ensure the efficiency of task selection by the worker.

Privacy, robustness, and scalability are also guaranteed for statistics collection, which will be discussed in Section V.

III. Optimization Model for Task Selection

In this section, we investigate fundamental trade-offs among three design goals and formulate two optimization problems to model them in the task selection component.

A. Definitions

Before proceeding further, we give the definitions for notations used in the rest of the paper as follows.

Definition 2. Contexts and Tasks

- Denote by $\mathcal{C} = \{c : c = 1, 2, \ldots, |\mathcal{C}|\}$ the set of all exact contexts. Each worker has an exact context c.
- Denote by $\hat{\mathcal{C}} = \{\hat{c} : \hat{c} = 1, 2, \ldots, |\hat{\mathcal{C}}|\}$ the set of all generalized contexts. Each exact context is mapped into a generalized context, and a generalized context may correspond to multiple detailed contexts.
- Denote by $\mathcal{T} = \{t : t = 1, 2, \ldots, |\mathcal{T}|\}$ the set of all tasks. For simplicity of notations, we treat tasks that have the same requirements for worker contexts and the same payment as one task. Each task may have multiple instances. The payment for successfully completing a task t is denoted as ρ_t.

Definition 3. Complete-and-Approve Rate (CAR): Both workers and the MC platform can earn some money when tasks are completed successfully (i.e., answers approved by task requesters). This can be characterized by the complete-and-approve rate (CAR), which can be calculated as N_1, the total number of workers with context c who have successfully completed task t, divided by N_2, the total number of workers with context c, i.e., $\mathrm{CAR}(t|c) = N_1/N_2$.

B. Trade-Offs among Utility, Privacy and Efficiency

The optimization model of task selection specifies how to choose tasks based on limited information about a worker. There are three conflicting design goals in this model: utility, privacy, and efficiency. These three goals cannot be optimized simultaneously. First, suppose that privacy and efficiency are optimized, which means that the worker provides no context about himself to the system and expects to receive a single task tailored for him. In this case, as long as the utility of tasks varies across different contexts, it is impossible for the recommendation server to choose a task that is of high utility for the worker. Second, consider the case that efficiency and utility are optimized. In order to find a task that has the highest utility for the worker, the recommendation server needs to know the exact worker context, compromising his privacy. Finally, if we want to ensure the optimality of utility and privacy, the recommendation server needs to recommend, without any prior knowledge of worker context, a set of tasks within which the worker can find one to maximize his utility. In this case, the efficiency becomes suboptimal since the recommended task set would be very large. If any of the above three goals is dropped, it is trivial to optimize the other two. Therefore, in practice, we have to find a good trade-off among these three goals.

C. Optimization Problem Formulation

In our framework, the worker first decides the amount of information about his private context to share with the server. Based on this limited information, the server selects L tasks $T \subset \mathcal{T}$ and sends them to the worker. Here, L determines the efficiency. Then the worker selects a task from the recommended L tasks, completes it, and returns the result back to the sever or task requester. Therefore, the task is selected jointly by the server and the worker in our framework.

As mentioned before, there are three conflicting goals. Although these goals cannot be optimized simultaneously, there are several candidate objective functions that optimizes the goals from different aspects. In the following, we choose an optimization objective function representing the utility and model the other two goals as constraints. In other words, we optimize the utility while allowing the worker to determine the efficiency and privacy requirements. Alternative objective functions are also discussed.

1) Computation at the worker side: Given a set of recommended tasks T, the worker selects one to complete. The behavior of the worker is supposed to be rational. In other words, the worker with exact context c would select the task that maximizes his own revenue, which can be modeled as

$$t^* = \underset{t \in T}{\operatorname{argmax}}\, \rho_t \cdot \mathrm{CAR}(t|c). \tag{2}$$

The computation of $\mathrm{CAR}(t|c)$ will be described later in this section.

2) Computation at the server side: Since the worker knows his own context, he can easily make the selection by maximizing his revenue. This is not true for the server as it can only select tasks based on the limited information provided by

the worker. To increase the relevance between recommended tasks and the worker, the server needs to recommend multiple tasks at the same time.

Assume that the server already has prior knowledge on the context-dependent click-and-approve rates, $\mathrm{CAR}(t|c)$, and the probability distribution over contexts. From the perspective of the server, its utility (i.e., revenue) depends on the task that the worker chooses, and we use the expected revenue of the set of tasks to quantify it. Since the server does not know the exact context c of the worker, it considers the probability of each of the exact contexts that generalize into $\hat{c}$ and calculates the expected revenue R of the set of tasks T as follows:

$$\mathbb{E}[R(T|\hat{c})] = \sum_{c:c\to\hat{c}} \Pr[c|\hat{c}] \cdot \alpha \cdot \max_{t\in T} \rho_t \cdot \mathrm{CAR}(t|c), \quad (3)$$

where α is the portion of revenue that the platform can obtain for each successful transaction. Let L denote the size of the task set. The server needs to select L tasks that maximize the expected revenue given a generalized context $\hat{c}$, i.e.,

$$T^* = \operatorname*{argmax}_{T\subseteq\mathcal{T}:|T|=L} \mathbb{E}[R(T|\hat{c})]. \quad (4)$$

3) Alternative Objectives: The above optimization model contains the extreme cases when task selection is taken solely at the server side ($L = 1$) or solely at the worker side ($L = |\mathcal{T}|$). For the former case, if the server recommends a single task based on a very generalized context provided by the worker, it is likely that the recommendation has a low utility. For the latter case, the server sends all the available tasks to the worker. The selection becomes inefficient, and the recommendation service is meaningless. Hence, the parameter L should be selected cautiously.

Instead of setting L as a predefined parameter, we can also include it as one of the design variables. This can be done by substituting $\mathbb{E}[R(T|\hat{c})] - \lambda \cdot L$ for the original objective $\mathbb{E}[R(T|\hat{c})]$ in (4), where λ is the weight of the efficiency metric L in the total objective function. As a result, the server selects a set of tasks that maximizes the new objective, i.e.,

$$T^* = \operatorname*{argmax}_{T\subseteq\mathcal{T}:|T|=L} \mathbb{E}[R(T|\hat{c})] - \lambda \cdot L. \quad (5)$$

In this way, the efficiency and the utility can be optimized jointly.

There are other options to model the utility as well. For example, we can incorporate the cost of a task into the objective such as time or other resources needed for completing a task. In this case, the selection process among a set of tasks for the worker becomes more complicated. A possible formulation might be $\max_{t\in T}(\rho_t - \mathrm{cost}_{t,c}) \cdot \mathrm{CAR}(t|c)$, where $\mathrm{cost}_{t,c}$ denotes the cost to complete task t by workers with context c. In addition, there might be a reservation wage w_r [24] below which the worker would not pick the task. Considering this, the process of task selection for a worker can be modeled as $\max_{t\in T}(\rho_t - \mathrm{cost}_{t,c}) \cdot \mathbf{1}_{\{\rho_t - \mathrm{cost}_{t,c} \geq w_r\}} \cdot \mathrm{CAR}(t|c)$.

IV. Solution Algorithms

In this section, we propose efficient solution algorithms for our optimization problem. In the following, we first consider the specific scenario which optimizes the objective of utility as in (3) and then discuss how to jointly optimize utility and efficiency as in (5). We mainly focus on computation at the server side, because the optimization problem at the worker side can be efficiently solved.

A. Approximation Algorithm for Optimizing the Utility

Both the server and the worker need to optimize their own objectives by solving (4) and (2), respectively. It is trivial for the worker to select the task from a set of L tasks, because L is usually designed to be a small number, and the optimization problem (2) can be directly solved efficiently. On the other hand, the server needs to select L tasks from the entire task space $\mathcal{T}$ according to (4). Directly solving this problem is computational intensive or infeasible. Actually, we have the following fact:

Proposition 1. *Given a generalized context $\hat{c}$, it is NP-hard to find a set of tasks T^* such that:*

$$T^* = \operatorname*{argmax}_{T\subseteq\mathcal{T}:|T|=L} \sum_{c:c\to\hat{c}} \Pr[c|\hat{c}] \cdot \alpha \cdot \max_{t\in T} \rho_t \cdot \mathrm{CAR}(t|c). \quad (6)$$

Proof. We can prove the NP-hardness by a reduction from the NP-hard maximum coverage problem. Details of this proof can be found in our technical report at [25]. □

Since the problem (6) is NP-hard, we design a greedy algorithm for the server as shown in Algorithm 1 below.

Algorithm 1 Greedy Algorithm for Profit Maximization

Input: $\mathcal{T}$, $\hat{c}$, L
Output: T
// initialization
$T \leftarrow \emptyset$;
repeat
  $t \leftarrow \operatorname{argmax}_{t\in\mathcal{T}} \mathbb{E}[R(T\cup t|\hat{c})] - \mathbb{E}[R(T|\hat{c})]$;
  $T \leftarrow T \cup \{t\}$;
until $|T| = L$
return T

By repeatedly choosing a task that maximizes the utility improvement, the greedy algorithm can be proved to approximate the optimal value within $1-1/e$, where e is the Euler's number (approximately 2.71828). Note that in [26], a greedy algorithm that solves the maximum coverage problem provides the same approximation ratio. However in their problem, the set either fully includes the element or not at all, while in our problem a task can partially matches the context, which complicates the problem and requires additional analysis. The proof of this approximation ratio for our approximation algorithm is given below.

Proposition 2. *The greedy algorithm approximates the optimal solution within a factor of $1 - 1/e$.*

Proof. Define a marginal utility function of adding set T' to T as follows:

$$f(T, T') = \mathbb{E}[R(T\cup T'|\hat{c})] - \mathbb{E}[R(T|\hat{c})].$$

The function $f(T, T')$ is submodular in the sense that $f(T_1, T') > f(T_2, T')$ for all sets $T_1 \subset T_2$. For $l = 1, 2, \ldots, L$, let $T_l = \{t_1, t_2, \ldots, t_l\}$ be the greedy solution constructed up to the end of the l-th stage; thus T_L is the final greedy solution returned. Similarly, let $T_L^* = \{t_1^*, t_2^*, \ldots, t_L^*\}$ be the optimal solution of any fixed order and $T_l^* = \{t_1^*, t_2^*, \ldots, t_l^*\}$ represents the first l tasks. Denote by $m(l) = \sum_{i=1}^{l} m_i$ the utility of T_l, where $m_l = f(t_l, T_{l-1})$ is the marginal utility by adding t_l to T_{l-1}. Similarly, denote by $m^*(l) = \sum_{i=1}^{l} m_i^*$ the utility of T_l^*, where $m_l^* = f(t_l^*, T_{l-1}^*)$. Our aim is to prove

$$m(L) \geq m^*(L) \cdot (1 - 1/e). \tag{7}$$

To this end, we first prove

$$m_l \geq (m^*(L) - m(l-1))/L, \forall l \in [1, L]. \tag{8}$$

The marginal utility of adding set T_L^* to set T_{l-1} is $f(T_{l-1}, T_L^*)$, which equals $\sum_{i=1}^{l} f(T_{l-1} \cup T_{i-1}^*, t_i^*)$. By the averaging argument, there exists an i such that $f(T_{l-1} \cup T_{i-1}^*, t_i^*\}) \geq (m^*(L) - m(l-1))/L$. We can then obtain $m_l = f(T_{l-1}, t_l) \geq f(T_{l-1}, t_i^*) \geq (m^*(L) - m(l-1))/L$, where the first inequality comes from how we choose t_l, and the second comes from submodularity.

We can then prove $m(l) \geq (1 - (1 - 1/L)^l) m^*(L), \forall l \in [1, L]$ by induction. When $l = 1$, the result holds: $m(l) = m_1 \geq m^*(L)/L = (1 - (1-1/L)^l) m^*(L)$ from (8). Suppose the inequality holds for l, i.e., $m(l) \geq (1 - (1-1/L)^l) m^*(L)$, we have

$$\begin{aligned} m(l+1) &= m(l) + m_{l+1} \geq m(l) + (m^*(L) - m(l))/L \\ &= m^*(L)/L + m(l)(1 - 1/L) \\ &\geq m^*(L)/L + m^*(L)(1 - (1-1/L)^l)(1 - 1/L) \\ &= (1 - (1 - 1/L)^{l+1}) m^*(L). \end{aligned}$$

Let $l = L$ in the above inequality, we have $m(L) \geq (1 - (1 - 1/L)^L) m^*(L) \geq (1 - 1/e) m^*(L)$, which completes the proof. □

B. Approximation Algorithm for Jointly Optimizing the Utility and Efficiency

As mentioned before, there are alternative objectives for the optimization problem. We now discuss how the server can jointly optimize the utility and efficiency in (5). As we show below, this is also an NP-hard problem.

Proposition 3. *Given a generalized context $\hat{c}$, it is NP-hard to find a set of tasks T^*, such that:*

$$T^* = \underset{T \subseteq \mathcal{T}: |T| = L}{\operatorname{argmax}} \sum_{c: c \to \hat{c}} \Pr[c|\hat{c}] \cdot \alpha \cdot \max_{t \in T} \rho_t \cdot \mathrm{CAR}(t|c) - \lambda \cdot L. \tag{9}$$

Proof. We can prove the NP-hardness of this problem by a reduction from Problem 1. Details of this proof can be found in our technical report at [25].

□

Below, we describe Algorithm 2 that approximately solves the above optimization problem (9) in polynomial time and give the analysis of approximation ratio in Proposition 4. In Algorithm 2, $L_{\max}$ denotes the maximum number of recommended tasks chosen by the worker beforehand.

Algorithm 2 Greedy Algorithm for Jointly Utility and Efficiency Optimization

Input: $\mathcal{T}$, $\hat{c}$, λ, $L_{\max}$
Output: T

```
// initialization
L ← 1, θ ← 0, T ← ∅;
while L ≤ L_max do
    Q ← ∅;
    repeat
        t ← argmax_{t∈T} 𝔼[R(Q ∪ t|ĉ)] − 𝔼[R(Q|ĉ)];
        Q ← Q ∪ {t};
    until |Q| = L
    if θ ≤ 𝔼[R(Q|ĉ)] − λ · L then
        θ ← 𝔼[R(Q|ĉ)] − λ · L;
        T ← Q;
    end if
    L ← L + 1;
end while
return T
```

Proposition 4. *The greedy algorithm approximates the optimal solution within a factor of $1 - 1/e$.*

Proof. Following the notations in the proof of Proposition 2, let $m(L)$ and $m^*(L)$ denote the objective function value for the greedy solution at a fixed L and the objective function value for the optimal solution at a fixed L, respectively. Denote by m_G and m_G^* the objective function value over all L for the greedy solution obtained by Algorithm 2 and the objective function value over all L for the optimal solution, respectively. Our aim is to prove $m_G \geq m_G^* \cdot (1 - 1/e)$.

Suppose that the optimal objective function value m_G^* is reached when $L = \tilde{L}$, we have $m^*(\tilde{L}) = m_G^*$. Now, from (7), $m(\tilde{L}) \geq m^*(\tilde{L}) \cdot (1 - 1/e) = m_G^* \cdot (1 - 1/e)$. Since $m_G \geq m(L)$, $\forall L = 1, \ldots, L_{\max}$, we have $m_G \geq m_{\tilde{L}} \geq m_G^* \cdot (1 - 1/e)$, which completes the proof. □

V. Privacy-Preserving Statistics Collection

In the previous sections, we have assumed that the server has prior information about worker statistics in the task selection component such as $\Pr[c|\hat{c}]$ and $\mathrm{CAR}(t|c)$. In this section, we describe how to obtain these statistics with privacy, robustness, and scalability guarantees.

A. Problem Overview

There are three parties in the offline statistics collection component: the MC server, workers, and a semi-honest third party (proxy). The server makes statistics queries and collects the results. Workers locally store their historical contexts as well as performance records, and answer queries. The proxy plays a mediation role between the server and the workers in order to protect worker privacy.

Threat Model and Assumptions. The server is assumed to be potentially malicious in the sense that it intends to violate

worker privacy. The server may attempt to use the statistics collection protocol to learn private information about workers, or deploy its own workers and manipulate their answers. Moreover, the server may also publish its collected worker statistics. Workers are also assumed to be potentially malicious in the sense that they may distort the final statistics learned by the server by submitting false or illegitimate answers. The proxy is assumed to be semi-honest or "honest-but-curious", which means it will faithfully follow the specified protocol, but may attempt to exploit additional information learned in executing the protocol. The proxy does not collude with other parties.

We assume that workers have correct public keys for the server and the proxy, that the server and the proxy have correct public keys for each other, and that all the corresponding private keys are securely kept. We also assume secure, reliable, and authenticated communication channels among the server, the proxy, and workers. Workers are assumed to be dynamic, which means that they may quit in the middle of the statistics collection process due to unstable wireless connection or power saving. Moreover, the computation and communication resources of worker devices are assumed to be limited.

In practice, as suggested in [16], the server may pay the proxy to execute the statistics collection protocol. Such a proxy has been used in previous papers [27], [28] and the relationship between the proxy and the MC server pre-exists in industry today which usually does not lead to collusion. For example, pharmaceutical companies pay an independent organization who evaluates the safety, quality, or performance of their products and may give unfavorable results against the pharmaceutical companies. Therefore, we believe that it is reasonable to have such a semi-honest proxy in our approach.

B. Computation of Worker Statistics Based on Counting

Based on a differentially private counting procedure, the statistics collection protocol gathers responses from workers and transforms the responses into statistics $\Pr[c|\hat{c}]$ and $\mathrm{CAR}(t|c)$. We first describe how these statistics can be computed based on a counting procedure. We will give the details of the counting procedure in Section V-C.

Calculating $\Pr[c|\hat{c}]$. The statistic $\Pr[c|\hat{c}]$ is calculated as the number of workers with context c divided by the number of workers with generalized context $\hat{c}$. Hence, the MC server should count the numbers of workers with context c and generalized context $\hat{c}$, respectively. To this end, the MC server constructs a statistics query which asks two questions: (1) "Is your private context c?" and (2) "Is your generalized context $\hat{c}$?". Both questions expect binary answers "yes" (represented by 1) or "no" (represented by 0). The answer from each worker k is a vector (b_k^1, b_k^2) that consists of two bits, each corresponding to a question. An example of the answer vector is shown in Fig. 3a. Therefore, given a privacy-preserving counting procedure, we can aggregate answers to these two questions from workers, and calculate $\Pr[c|\hat{c}]$ in a privacy-preserving manner.

Calculating $\mathrm{CAR}(t|c)$. The statistic $\mathrm{CAR}(t|c)$, as defined in Section III-A, is calculated as the total number of workers with context c who have completed task t divided by the total number of workers with context c. The MC server also generates a query that consists of two questions: (1) "Is your context c?" and (2) "If your context is c, have you successfully completed task t?". The answer to these two questions is contained in a two-bit vector as well. If the context of the worker is not c, the answer of the worker would be $[0, 0]$; if the context of the worker is c, and he has successfully completed the task, his answer would be $[1, 1]$; if the context of the worker is c but he does not complete the task, his answer would be $[0, 1]$. Here, the first bit of the answer vector indicates that the worker satisfies both context c and completion of task t, while the second bit indicates whether the worker's context is c as shown in Fig. 3a. Similarly, we can compute $\mathrm{CAR}(t|c)$ using a counting procedure over answers to these questions from workers.

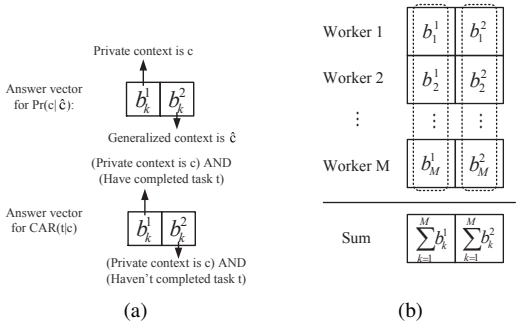


Fig. 3: (a) Illustration of the answer vector for worker k; (b) Aggregation process of answer vectors from M workers.

In practice, the MC server first sets M and ϵ, where M indicates the number of workers that need to be queried and ϵ is the privacy budget that controls the amount of noise. The queries and the parameters M and ϵ are then broadcasted to workers, whose answers will be added bit by bit as shown in Fig. 3b by a privacy-preserving counting procedure explained below.

C. Distributed Differentially-Private Counting Procedure

We now describe our differentially private counting procedure which is the key part of the statistical collection approach. The counting procedure takes answers from workers as the input data, and outputs a noisy sum with differential privacy guarantees, i.e., a sum that does not significant change with the presence or absence of a single worker. Since in our distributed setting, the data are owned by workers themselves, it is non-trivial to add the differential noise to the distributed data. There are a few works which provide differential privacy in a distributed setting [23], [29], [30]. However, they either have a high computation cost on each user [23] or requires users to be online during the whole computation process [29], [30], rendering them impractical for a large-scale setting as our scenario.

To ensure the scalability of our approach, we adapt the protocol in [16], which employs a semi-honest proxy to achieve differential privacy under distributed setting for a

different application scenario. The proxy aggregates answers from workers and adds noise to the sum; however, it is unable to learn the value of answers or their sum. Moreover, the proxy adds noise "blindly" such that it does not know the value of the noise. In this way, the proxy is unable to recover the accurate count by subtracting the noise from the published final statistics.

For ease of presentation, we call each encrypted binary bit as a *coin*, and call a set of coins as a *bucket*. We summarize the counting protocol as follows: **Step 1:** The server formulates a query request for a specific statistics and specifies the number of queried workers M and the privacy parameter ϵ for this query. **Step 2:** The proxy sends the query to qualified workers. **Step 3:** After a worker receives the query, he constructs an answer vector, encrypts his answer bits with the public key of the MC server, and sends the ciphertexts (i.e., coins) to the proxy. **Step 4:** The proxy aggregates the coins into buckets and adds blind binomial noise N based on ϵ in each bucket. **Step 5:** The proxy forwards the answers to the MC server. **Step 6:** The MC server decrypts each encrypted binary answer with its private key, sums up the decrypted values in each bucket, and subtracts $N/2$ from the sum in order to cancel the added noise. Since the MC server cannot tell who constructs encrypted answers, the identities of workers are anonymized. For detailed steps of the protocol, readers may refer to [16], which has similar information flow as our protocol.

D. Noise Addition

The amount of noise required to achieve (ϵ, δ)-differential privacy is calculated in [23] and described below.

Let N be the number of unbiased coins added in a bucket, i.e., the amount of Binomial noise. The statistics collection algorithm achieves (ϵ, δ)-differential privacy if $N \geq 64\ln(2/\delta)/\epsilon^2$, where parameters δ and M are selected by the server. Suppose that any query of each person is sensitive, then $\delta > 1/M$ indicates the disclosure of at least one person's privacy. Therefore, δ is selected to be smaller than $1/M$. With this constraint, the amount of noise added should satisfy

$$N \geq \frac{64\ln(2M)}{\epsilon^2}. \tag{10}$$

The semi-trusted proxy should collaborate with workers to generate unbiased and blind coins so that neither party can determine or know the amount of added noise. To this end, coins are first generated by workers and then "flipped" by the proxy. Coin-flipping can be realized by the XOR-homomorphic encryption, where the ciphertext of the XOR of two binary values is equal to the product of their ciphertexts, i.e., for any $b, b' \in \{0, 1\}$, we have $e(b) \cdot e(b') = e(b \oplus b')$, where $e(\cdot)$ is the encryption operator. With this homomorphic property, two parties can collaboratively generate an encrypted value of either 0 or 1 while no single party can know or control the final results. As long as one of the two parties is unbiased, the final results would be unbiased. We use the Goldwasser-Micali (GM) cryptosystem [31] for coin generation, which has the desired XOR-homomorphic property, and is also very efficient for encrypting binary values.

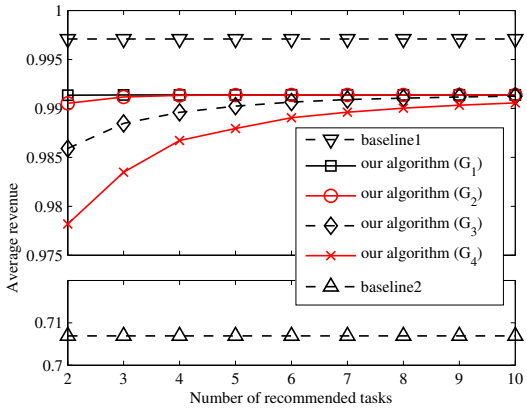


Fig. 4: Effect of generalization level on the average revenue in Problem (4).

E. Properties of the Statistics Collection Approach

The aforementioned statistics collection approach provides the following three properties: (i) *Privacy.* Our approach ensures differential privacy for all workers. Whenever a worker participates in the statistics collection procedure, he reveals some information about himself. Such kind of privacy loss is quantified by the privacy budget [23], [32]. The privacy loss is accumulated across queries until it surpasses the worker's privacy budget. Then the worker stops contributing any data in the statistics collection procedure. This provides the best privacy for the worker. (ii) *Scalability.* By scalability, we refer to low per-worker computation cost and resistance to worker dynamics. In our approach, the cost per worker is $\mathcal{O}(1)$. Hence even when the number of workers is large, the cost for individual worker does not change much. Moreover, in our approach, workers only need to submit answers once and no further communication is required after that. Therefore, our approach allows workers to leave after they submit their answers. This is important when the number of workers is large because it is difficult to keep all workers online during the whole statistics collection process. (iii) *Robustness.* With the GM encryption, we are able to bound the error brought by malicious workers because a malicious worker would be unable to distort the final sum by more than 1. The result submitted by each user can only be 0 or 1 and other illegitimate values can be easily detected by checking the Jacobi symbols of ciphertexts at the proxy. Suppose 1% of workers are malicious, the error introduced by malicious worker would be less than 1%.

VI. Performance Evaluation

To evaluate the performance of the proposed optimization algorithms, we generate a synthetic dataset to simulate the statistics $\Pr(c)$ and $\mathrm{CAR}(t|c)$. Without loss of generality, we assume the frequency of worker contexts is uniformly distributed. The data set includes 2048 exact contexts and 10000 different tasks. The detailed contexts can be generalized at four different levels. There are 512 level-1 generalized contexts denoted as "G_1", 128 level-2 generalized contexts

TABLE I: Running time (in unit of seconds) of recommending L tasks from a set of 100 tasks ($L = 2$ or 3)

L	G_1		G_2		G_3		G_4	
	Optimal	Greedy	Optimal	Greedy	Optimal	Greedy	Optimal	Greedy
2	13.0	8.1	13.3	2.0	14.0	0.1	24.9	0.04
3	1125.2	15.7	1267.4	3.8	1275.6	0.3	1838.1	0.1

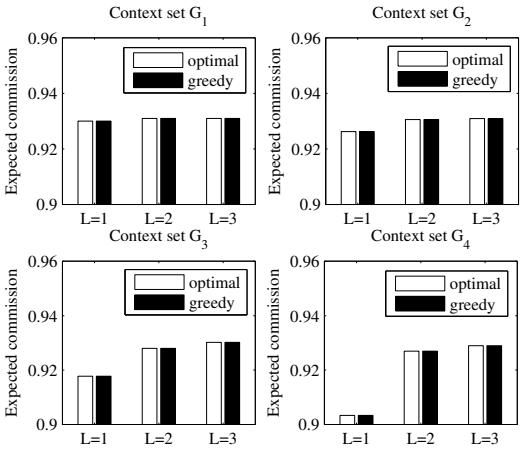


Fig. 5: Performance of Approximation Algorithm 1.

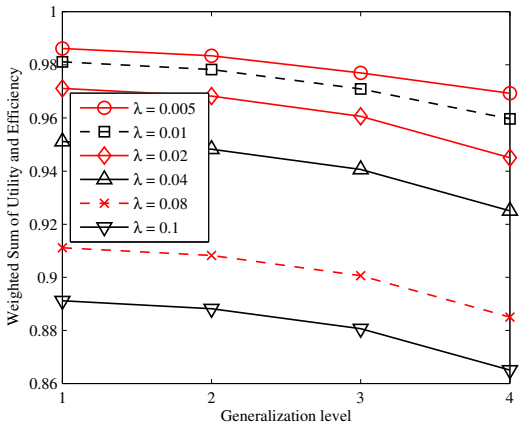


Fig. 6: Effect of the generalization level and the weight for efficiency in Problem (5).

denoted as "G_2", 8 level-3 generalized contexts denoted as "G_3", and 2 level-4 generalized contexts denoted as "G_4". The statistic $\mathrm{CAR}(t|c)$ is generated in a way such that the closer two exact contexts are, the more similar the distributions of $\mathrm{CAR}(t|c)$ would be. The $\mathrm{CAR}(t|c)$ of a task t for workers with the same exact context c follows a uniform distribution. The payments of tasks are set as a random value between 0 and 10, and the ratio of commission α is chosen to be 0.1.

Firstly, we test the effectiveness of the task recommendation model. To this end, we compare our proposed algorithm (Algorithm 1) with two baseline algorithms, "baseline1" and "baseline2". The first baseline algorithm uses the exact worker context as the input. With the exact worker context, the algorithm directly chooses the task that maximizes the revenue gained by the MC platform. Worker privacy is compromised in this algorithm to trade for utility and efficiency. On the contrary, in the second baseline algorithm, no context information is used, and therefore worker privacy is maximized. This algorithm does not consider the difference of worker contexts and recommends tasks that have highest payments.

Fig. 4 shows the expected revenue of the MC platform by adjusting the size of the recommended task set L. We run the experiments using six different algorithms, including two baseline algorithms and Algorithm 1 with four different levels of generalized contexts. Intuitively, the two baseline algorithms serve as a upper bound and a lower bound of other algorithms, respectively, which is clearly shown in the figure. The expected revenue of Algorithm 1 increases when more context information is used. For a specific level of generalization, the expected revenue increases with L. For example, when the generalization level is 3 (which corresponds to "G_3" in the figure), the revenue increases from 0.986 to 0.991 as L increases from 2 to 10. Note that the performances of the two baseline algorithms do not change with L because they always select the task that maximizes the expected revenue regardless of L. We can see that when privacy level increase from G_1 to G_4, the decrease in the average revenue is not significant, this shows the effectiveness of our privacy preserving approach.

Secondly, we evaluate the performance of the proposed approximation algorithms. Due to the NP-hardness of the original optimization problem, the optimal solution becomes intractable in practice when either L or the task space is large. Therefore, we use a reduced size of data set for this experiment (i.e., 100 tasks and $L = 1, 2, 3$). Fig. 5 compares the performances of Algorithm 1 and the optimal algorithm. We see that there is little difference between the two algorithms for $L = 1, 2, 3$ and $|\mathcal{T}| = 100$. The difference between the two algorithms may grow as L becomes larger, but we have proved in previous sections that our approximation algorithm has an approximation ratio of $1 - 1/e$.

Thirdly, we show the performance of Algorithm 2, which jointly optimizes utility and efficiency. Fig. 6 plots the weighted sum of utility and efficiency with the weight coefficient λ ranging from 0.005 to 0.1. For each λ, the x-axis represents the level of context generalization, and the y-axis represents the the weighted sum of utility and efficiency. Same as what we get from Fig. 4, the weighted sum decreases as the level of generalization increases, which shows a clear trade-off between utility and privacy. With the increase of λ, the optimized weighted sum decreases. This is reasonable because it is shown in (5) that for the same list of recommended tasks, the weighted sum decreases with the increase of λ. As a result, the optimal weighted sum is expected to decrease as well.

VII. System Overhead

In this section, we analyze the system overhead of the proposed framework, including both task selection and statistics collection components.

We list the estimated running time for the task selection component in Table I. Since the time for the optimal algorithm grows exponentially as L grows, we only run this algorithm at $L = 2$ or 3 with a small dataset where the number of tasks is 100. We can see that the time to get an optimal solution grows rapidly with L, while the time for the proposed greedy algorithm is linear with respect to L.

In the following, we analyze the computation, storage and communication overhead for statistics collection. Firstly, we analyze the computation overhead for the GM cryptosystem. With a 1024-bit key length, a smartphone running Android 2.2 with 1GHz processor can execute more than 800 encryptions within one second [16]. Since workers only need to execute the encryption process once for each query request, the computation cost is negligible for them. The proxy is implemented with Apache Tomcat 6.0.33, which can execute more than $15,000$ GM encryptions, or $123,000$ homomorphic XORs per second, and the server is implemented with Java source code, which can execute more than 6000 GM decryptions per second. Consider a normal setting where there are 5000 workers with 100 different exact contexts which generalize to the same generalized context, and there are 90 tasks relevant to this generalized context. Suppose 10% of the workers participate in the statistics collection process, the proxy needs to execute 18 encryptions and 18 homomorphic XORs for a single statistic query when the privacy parameter ϵ is set to 5 according to (10). In order to calculate all the statistics needed for the task selection model, the proxy needs to execute 18×27200 encryptions and 18×27200 homomorphic XORs, which takes 31 seconds and 4 seconds, respectively. For the same setting, the server needs to decrypt a total of $(500 + 18) \times 27200$ coins, which takes 36 minutes. Note that the statistics can be calculated offline and are reusable among workers with similar contexts. By contrast, if the approach is implemented with the Paillier system, in order to calculate statistics for a task selection model, it takes the mobile worker, the proxy, and the server 4 seconds, 139 minutes, and 4500 minutes, respectively. Therefore, the GM crytosystem use in our framework is highly efficient.

Next, we discuss the storage and communication bandwidth requirements. Since a worker transmits no more than 3 coins for each statistics collection query and a periodically generated coin for noise addition, the storage requirement for him is in the order of kB. Considering that workers can selectly respond to the requests, the storage overhead is quite acceptable. Suppose the coins should be sent out within one second, the bandwidth requirement would be around 1 kB/s. As for the proxy, since it needs to store all queried coins and noise coins before sending them to the server, which is about 518×27200 coins in total in the above setting, the storage overhead would be about 1.7 GB. Since the statistics collection process are computed beforehand, we assume the maximum transmission time is 30 minutes. Therefore, the bandwidth for sending these data is 1 MB/s. Note that although the storage requirement for computing a statistic is not small, in practice, the statistic only needs to be computed once and updated at a low frequency after it has been calculated. The overheads for the proxy to update the statistics are at the same order of the overhead for workers.

VIII. Related Work

In this section, we review some works related to our problem in the literature.

Previous works on privacy issues of mobile applications mainly focus on location privacy in location-based services, and they use either obfuscation to hide true locations [33], [34] or aggregation to hide individual sensitive information [35]. However, none of them discuss how to recommend tasks in the absence of accurate private information. In this paper, we consider the fundamental trade-offs among privacy, utility, and efficiency, and provides a flexible framework to select tasks at different trade-off points.

There are a few works in task recommendation for crowdsourcing applications. Ho and Vaughan [36] address the scenario where heterogeneous tasks are assigned to workers with unknown skill sets with an exploration-exploitation trade-off. Yuen et al. [37] utilize performance history and task search history to model user preference and recommend tasks for a user based on his/her preference. Ambati et al. [38] implicitly model user skills and interests, and recommend tasks based on user preference. However, these works have not addressed the specific privacy concerns in MC scenarios where tasks should be recommended to workers based on private, sensitive information. To et al. [39] consider spatial crowdsourcing where the cost for a task depends on the distance between the worker and the task and implement a toolbox for privacy-preserving spatial crowdsourcing. Pournajaf et al. [40] formulate an optimization problem to minimize the cost for all workers in spatial crowdsourcing. These previous work mainly focus on coordinated task assignment where the crowdsourcing server decides which task is completed by a worker, whereas our task recommendation scenario is autonomous task selection that let workers select tasks from a list of tasks by themselves.

IX. Conclusion

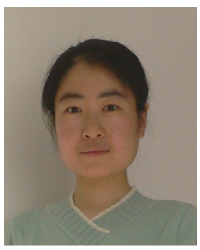
We have considered the privacy issues in task recommendation for mobile crowdsourcing. We have proposed a task recommendation framework which recommends mobile crowdsourcing tasks without violating worker privacy. The proposed framework is comprised of two components: task selection component and statistics collection component. In the task selection component, we have developed a privacy-aware optimization model of task selection that considers the intrinsic trade-offs among utility, privacy and efficiency and selects tasks based on the limited information of worker context. Workers have the choice of how much private information they are willing to share with the server. In the statistics collection component, we have adopted an approach that gathers necessary statistics about worker contexts while guaranteeing differential privacy. We have evaluated the effectiveness and

efficiency of the proposed framework and analyzed the system overhead. For future work, we intend to incorporate other popular task recommendation algorithms such as collaborative filtering. We also plan to jointly consider task assignment and task recommendation problems in MC systems.

REFERENCES

[1] Waze. [Online]. Available: https://www.waze.com
[2] Uber. [Online]. Available: https://www.uber.com/
[3] Stereopublic. [Online]. Available: http://www.stereopublic.net/
[4] OpenSignal. [Online]. Available: http://opensignal.com/
[5] C. Freifeld, R. Chunara, S. Mekaru, E. Chan, T. Kass-Hout, J. Brownstein *et al.*, "Participatory epidemiology: use of mobile phones for community-based health reporting," *PLoS medicine*, vol. 7, no. 12, pp. e1 000 376–e1 000 376, 2009.
[6] Y. Chon, N. D. Lane, Y. Kim, F. Zhao, and H. Cha, "A large-scale study of mobile crowdsourcing with smartphones for urban sensing applications," in *UbiComp*, 2013.
[7] A. Tamilin, I. Carreras, E. Ssebaggala, A. Opira, and N. Conci, "Context-aware mobile crowdsourcing," in *ACM UbiComp*, 2012.
[8] Y. Wang, Y. Huang, and C. Louis, "Respecting user privacy in mobile crowdsourcing," *SCIENCE*, vol. 2, no. 2, pp. pp–50, 2013.
[9] X. Shen, B. Tan, and C. Zhai, "Implicit user modeling for personalized search," in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005.
[10] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *VLDB*. VLDB Endowment, 2006, pp. 763–774.
[11] S. Guha, A. Reznichenko, K. Tang, H. Haddadi, and P. Francis, "Serving ads from localhost for performance, privacy, and profit," in *HotNets*, 2009.
[12] M. Fredrikson and B. Livshits, "Repriv: Re-imagining content personalization and in-browser privacy," in *S&P*. IEEE, 2011, pp. 131–146.
[13] S. Chakraborty, K. R. Raghavan, M. P. Johnson, and M. B. Srivastava, "A framework for context-aware privacy of sensor data on mobile systems," in *HotMobile*, 2013.
[14] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, M. K. Reiter *et al.*, "Playing devil's advocate: Inferring sensitive information from anonymized network traces." in *NDSS*, 2007.
[15] B. F. Ribeiro, W. Chen, G. Miklau, and D. F. Towsley, "Analyzing privacy in enterprise packet trace anonymization." in *NDSS*, 2008.
[16] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, "Towards statistical queries over distributed private user data," in *NSDI*, 2012.
[17] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-enhancing personalized web search," in *ACM WWW*, 2007.
[18] C. S. Jensen, H. Lu, and M. L. Yiu, "Location privacy techniques in client-server architectures," in *Privacy in location-based applications*. Springer, 2009, pp. 31–58.
[19] E. Baralis, L. Cagliero, T. Cerquitelli, P. Garza, and M. Marchetti, "Context-aware user and service profiling by means of generalized association rules," in *Proceedings of the 13th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems: Part II*. Springer-Verlag, 2009, pp. 50–57.
[20] J. Shi, Y. Zhang, and Y. Liu, "Prisense: privacy-preserving data aggregation in people-centric urban sensing systems," in *IEEE INFOCOM*, 2010.
[21] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "Poolview: stream privacy for grassroots participatory sensing," in *ACM SenSys*, 2008.
[22] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Security and Trust Management*. Springer, 2011, pp. 226–238.
[23] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *EUROCRYPT*, 2006.
[24] J. J. Horton and L. B. Chilton, "The labor economics of paid crowdsourcing," in *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 2010, pp. 209–218.
[25] Y. Gong, Y. F. Fang, and Y. Guo, "Optimal task recommendation for mobile crowdsourcing with privacy control." [Online]. Available: http://plaza.ufl.edu/ymgong/PrivacyMC_report.pdf
[26] D. S. Hochbaum and A. Pathria, "Analysis of the greedy approach in problems of maximum k-coverage," *Naval Research Logistics*, vol. 45, no. 6, pp. 615–627, 1998.
[27] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *IEEE INFOCOM*, 2010.
[28] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *IEEE INFOCOM*, 2010.
[29] E. Shi, T.-H. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *NDSS*, 2011.
[30] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *ACM SIGMOD*, 2010.
[31] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the fourteenth annual ACM symposium on Theory of computing*. ACM, 1982, pp. 365–377.
[32] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.
[33] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *PERVASIVE*, 2005.
[34] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *ACM CCS*, 2012.
[35] J. W. Brown, O. Ohrimenko, and R. Tamassia, "Haze: Privacy-preserving real-time traffic statistics," in *ACM SIGSPATIAL*, 2013.
[36] C.-J. Ho and J. W. Vaughan, "Online task assignment in crowdsourcing markets," in *AAAI Conference on Artificial Intelligence*, 2012.
[37] M.-C. Yuen, I. King, and K.-S. Leung, "Task recommendation in crowdsourcing systems," in *ACM CrowdKDD*, 2012.
[38] V. Ambati, S. Vogel, and J. Carbonell, "Towards task recommendation in micro-task markets," in *Proceedings of The 25th AAAI Workshop in Human Computation*, 2011.
[39] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," in *VLDB*, 2014.
[40] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *IEEE MDM*, 2014.

Yanmin Gong (S'10) received the B.Eng. degree in electronics and information engineering from Huazhong University of Science and Technology, China, and the M.S. degree in electrical engineering from Tsinghua University, China, in 2009 and 2012, respectively. She is currently a PhD student in electrical and computer engineering at the University of Florida. Her research interests include cyber-security and privacy.

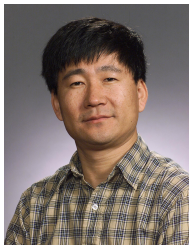


Lingbo Wei (M'15) received the B.S. degree in Mathematics from Shaanxi Normal University in 2001, the M.S. degree in Cryptography from Xidian University in 2005, and the Ph.D. degree in Information Security from the Institute of Software, Chinese Academy of Sciences in 2009. From June 2009 to October 2011, she was a Postdoctoral Fellow at Beihang University, and from November 2011 to October 2014, she was a Postdoctoral Fellow at Shanghai Jiao Tong University. She joined the University of Science and Technology of China in November 2014 as an Associate Professor of the School of Information Science and Technology. Her research interests include network security, privacy protection, and applied cryptography.

Yuanxiong Guo (M'14) received the B.Eng. degree in electronics and information engineering from Huazhong University of Science and Technology, China, in 2009 and the M.S. degree and Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2012 and 2014, respectively.

He is now an Assistant Professor in the School of Electrical and Computer Engineering at Oklahoma State University, Stillwater, OK, USA. His research interests include smart grids, cyber-physical systems, sustainable computing and networking, and critical infrastructure cybersecurity and resilience. He is the recipient of the Best Paper Award in IEEE Global Communications Conference 2011.

Chi Zhang (M'11) received the B.E. and M.E. degrees in Electrical and Information Engineering from Huazhong University of Science and Technology, China, in 1999 and 2002, respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Florida in 2011. He joined the University of Science and Technology of China in September 2011 as an Associate Professor of the School of Information Science and Technology. His research interests are in the areas of network protocol design and performance analysis, and network security particularly for wireless networks and social networks. He has published over 60 papers in journals such as IEEE/ACM Transactions on Networking, IEEE Journal on Selected Areas in Communications, and IEEE Transactions on Mobile Computing and in networking conferences such as IEEE INFOCOM, ICNP, and ICDCS. He has served as the Technical Program Committee (TPC) members for several conferences including IEEE INFOCOM, ICC, GLOBECOM, WCNC and PIMRC. He is the recipient of the 7th IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award.



Yuguang Fang (F'08) received MS degree from Qufu Normal University, China, in 1987, and Ph.D. degrees from both Case Western Reserve University and Boston University in 1994 and 1997, respectively. He joined the Department of Electrical and Computer Engineering at University of Florida since 2000. Dr. Fang received the US NSF Faculty Early Career Award in 2001 and the US ONR Young Investigator Award in 2002, and is a recipient of the Best Paper Award in IEEE International Conference on Network Protocols in 2006. He also received a 2010-2011 UF Doctoral Dissertation Advisor/Mentoring Award and IEEE Communications Society WTC Recognition Award. He served as the Editor-in-Chief of IEEE Wireless Communications and is currently serving as the Editor-in-Chief of IEEE Transactions on Vehicular Technology. He is a Fellow of IEEE.