A Prediction-based User Selection Framework for Heterogeneous Mobile CrowdSensing

Yongjian Yang, Wenbin Liu, En Wang*, Jie Wu, *Fellow, IEEE*

Abstract—Mobile CrowdSensing is a new paradigm in which requesters launch tasks to the mobile users who provide the sensing services. The tasks, in practice, are usually heterogeneous (have diverse spatial-temporal requirements), which make it hard to select an efficient subset of users to perform the tasks. In this paper, we present a point of interest (PoI) based mobility prediction model to obtain the probabilities that tasks would be completed by users. Based on it, we propose a greedy offline algorithm to select a set of users under a participant number constraint. Furthermore, we extend the user selection problem to a more realistic online setting where users come in real time and we decide to select or not immediately. We formulate the problem as a submodular k-secretaries problem and propose an online algorithm. Finally, we design a distributed user selection framework *Crowd UserS* and implement an Android prototype system as proof of the concept. Extensive simulations have been conducted on three real-life mobile traces and the results prove the efficiency of our proposed framework.

Index Terms—Mobile CrowdSensing, User Selection, Mobility Prediction, Submodular k-Secretaries Problem.

1 Introduction

Mobile CrowdSensing (MCS) [2], a novel sensing mechanism that has been presented in recent years, it serves the vital purpose of exploiting the ubiquitous mobile devices carried by users in order to provide complex computation and sensing services. Unlike traditional sensing methods which rely on the static sensors or specialized monitoring stations (even need the dedicated staffs), in MCS, any human can perform the sensing tasks at various times and places, with the help of their mobile devices represented by smartphones. Nowadays, many urban tasks, such as environment and traffic monitoring, could be addressed perfectly by using MCS [3].

MCS would provide great convenience services, when it has the effective users. Obviously, user selection is the foundation of MCS and there has been so much research on it [4–13]. In these works, researchers mainly focused on the user selection over opportunistic networking [4, 7], and the similar scenes like piggyback [10], vehicle-based [8], and self-organized MCS [5] also had been proposed. Recently, the realistic settings on the sensing tasks, including deadlines [6], multi-task [11] and heterogeneous sensing tasks [12], have been further studied. In this paper, we focus on the sensing tasks which are heterogeneous in terms of spatial-temporal dimensions in MCS, called Heterogeneous MCS (H-MCS). Specifically, the spatial-temporal-sensitive tasks can have different spacial and temporal requirements and various sensing periods, and users need to perform the sensing tasks within the required times and locations. An example (Fig. 1) illustrates the generality of H-MCS: The MCS server wants to know the queuing situation at 12:00 in the restaurant. Thus, it needs to recruit the users who will reach the restaurant at 12:00. While the users who arrive at any other time are of no help. Therefore, a higher request is made for the user selection of H-MCS.

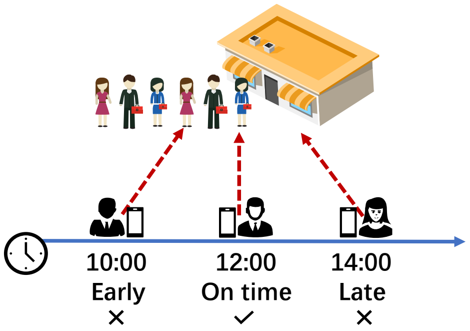


Fig. 1: An example of H-MCS: The requester wants to know the queuing situation at 12:00. The users who will arrive at any other time are of no help.

To solve this problem, we mainly face two challenges: how to measure which user is better and how to select a suitable user set. For the first challenge, we consider that the users who would satisfy the spatial-temporal requirements are the better ones, that is, we recruit the users who can reach the locations at required times. However, previous studies do not consider the temporal and spatial requirements adequately (*e.g.*, [4–9]). Some studies present the user selection algorithms based on the known and predetermined user mobility (*e.g.*, [4, 9]), and some researches have tried to use the predicted or probabilistic trajectories[4–6, 11, 12]. However, these predicted or probabilistic trajectories can hardly be obtained and applied well in practical settings. On the one hand, the fine-grained mobility prediction

- *Yongjian Yang, Wenbin Liu and En Wang are with the Department of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China.*
(e-mail: yyj@jlu.edu.cn; liuwb16@mails.jlu.edu.cn; wangen@jlu.edu.cn)
- *Jie Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA.*
(e-mail: jiewu@temple.edu)

(Corresponding author: En Wang.)

A preliminary version of this work appeared in the Proceedings of the IEEE International Conference on Sensing, Communication and Networking (SECON 2017) [1].

is difficult, especially for the large-scale MCS applications. On the other hand, it would cost a lot on personalized mobility prediction, especially for a large number of users. Additionally, the privacy issues also need to be noticed.

Even after we obtain the trajectories, we still face the second challenge: how to select a suitable user set? Most of previous studies use the greedy heuristics to select users based on their utility functions, while these works are typically used in offline scenario, *i.e.*, selecting users with the global information. We believe that the online scenario where we make the decisions without the benefit of future information, is the most relevant to MCS applications. In online case, tasks and users would be coming in real time and we must decide whether to select the user or not immediately, which is so difficult to deal with.

In this paper, we focus on the two challenges above, and propose a user selection framework for H-MCS, called *Crowd UserS*. We first utilize a simplified but effective point of interest (PoI) based prediction model to make the mobility prediction with better precision and less computation, and propose a greedy offline user selection algorithm. Then, we extend the problem to the online setting and propose an online algorithm to make it more practical. Finally, we present the distributed user selection framework *Crowd UserS* and implement a prototype system.

For mobility prediction, we first simplify the mobility prediction to PoI based prediction, by using the semi-Markov mobility prediction model, which uses landmark trajectory prediction to determine the probability distribution of the user's arriving at some landmarks for each time unit [1, 14, 15]. Using this prediction model, we can ignore the trajectories at other places but focus on the small areas containing the locations of sensing tasks, *i.e.*, PoI. Thus, we only consider the users' movements among PoIs and make predictions on PoIs. This prediction model helps us obtain the probabilistic results with better precision and less computation. Then, we consider whether the user could perform sensing task on time as a probabilistic problem. Moreover, different from previous studies [5, 6, 8, 9, 11, 12, 16], we further take the uploading problem into consideration and present two uploading ways (cellular links and collection points) for two common kinds of tasks (time-sensitive tasks and delay-tolerant tasks) [16]. For time-sensitive tasks, users need to upload the sensing data immediately after performing sensing tasks, thus they should use the cellular links. For delay-tolerant tasks, users can hold and upload the sensing data before the deadline, thus they would like to use free collection points, such as WiFi and Roadside APs, in order to reduce their costs. In summary, with the help of PoI based prediction model, we obtain the probabilities that the users complete (perform and upload) the spatial-temporal-sensitive sensing tasks.

For user selection, we would like to select a set of users based on the mobility prediction. The selected users would collaboratively perform sensing tasks as many as possible under a participant number constraint. Actually, the user selection problem is an NP-hard problem. We design a greedy offline algorithm to solve it, with a competitive ratio of $1 - \frac{f(u_{max})}{f(\mu^*)-|\mu|}$. Furthermore, we extend the problem to a more general online scenario, where users come in real time and we decide to select or not immediately. We cast the dynamic user selection problem as a variant of the secretary problem and propose an online algorithm to select users by stages, with an approximation ratio of $\frac{1-1/e}{7}$.

Additionally, combining the mobility prediction and user selection algorithms, we present a distributed framework, called *Crowd UserS*, in which users could make their own predictions and only need to return the probabilities. This distributed user selection framework not only reduces the centralized computing but also protects privacy up to a certain point. Finally, we also implement a prototype system to evaluate the performance.

The main contributions of this paper are briefly summarized as follows:

- We present a Heterogeneous Mobile CrowdSensing system model in which the tasks have diverse spatial-temporal requirements. We simplify the mobility prediction to point-of-interest (PoI) based prediction, and obtain the probabilities that the spatial-temporal-sensitive tasks will be completed on time.
- We formulate a new prediction based user selection problem with a participant number constraint and prove the NP-hardness. We propose an offline greedy user selection algorithm, with an approximation ratio of $1-\frac{f(u_{max})}{f(\mu^*)-|\mu|}$. Furthermore, we extend our problem to a more realistic scenario where the tasks and users would be coming in real time and we must decide whether to select the user or not immediately. We propose an online algorithm to deal with it, with an approximation ratio of $\frac{1-1/e}{7}$.
- We propose *Crowd UserS*, a distributed user selection framework for H-MCS, which reduces the centralized computing and partially protects privacy by using the distributed storage and computing architectures. In addition, we implement a prototype system on the Android platform.
- We evaluate the proposed algorithms on three real-life traces and the results show that our algorithms achieve better performances than the statistical and random selection strategy, even have good enough performances compared with the strategy with known user mobility.

The remainder of the paper is organized as follows. Firstly, we review related works in Section 2. Then, the system model and the problem formulation are introduced in Section 3. In Section 4, we focus on the PoI based mobility prediction. The offline/online user selection algorithms are proposed in Section 5 and 6, respectively. The framework and prototype system will be shown in Section 7. Finally, the performance is evaluated through extensive simulations in Section 8, and we discuss and conclude this paper in Section 9 and 10.

2 RELATED WORKS

2.1 User Selection

Recently, many researchers take part in the study of user selection in Mobile CrowdSensing. Merkouris Karaliopoulos *et al.* [4] use opportunistic networking techniques to address the user selection problem and formulate the problem as instances of the minimum cost set cover problem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2879098, IEEE Transactions on Mobile Computing

Furthermore, they prove the NP-hardness of the problem and propose practical greedy heuristics. Zongjian He $et\ al.$ [8] propose a greedy approximation algorithm and a genetic algorithm for the user selection problem in vehicular networks based on the predicted trajectories. Lingjun Pu $et\ al.$ [5] propose a novel framework called Crowdlet for self-organized mobile crowdsourcing and formulate an online multiple stopping problem formulation to dynamically select better users. Mingjun Xiao $et\ al.$ [6] further study the deadlines of tasks and design a submodular utility function. Moreover, they extend the user selection problem to the case where the sensing duration is taken into consideration, and propose the approximation algorithm. In these works, researchers focus on the locations but ignore the complex but practical temporal requirements of tasks ($e.g.$, beginning and ending time). Therefore, the existing methods cannot deal with our problem as the user selection algorithm in H-MCS is quite different.

Among the existing works, some works notice the various spatial-temporal requirements and sensing periods. Hanshang Li $et\ al.$ [12] focus on a new dynamic selection problem for spatial-temporal-sensitive mobile crowdsensing tasks, with a goal of minimizing the sensing cost while satisfying certain levels of coverage. In this work, sensing tasks can arrive at any time and may have various temporal/spatial requirements and with various sensing periods. They formulate the dynamic participant recruitment problem with spatial-temporal-sensitive sensing tasks in a large-scale piggyback Mobile CrowdSensing system and propose three greedy algorithms (one offline and two online) to tackle it. Yan Liu $et\ al.$ [11] propose two greedy-enhanced genetic algorithms to deal with the multi-task user selection problem for time-sensitive tasks and delay-tolerant tasks respectively. For time-sensitive tasks, it performs Participatory Sensing and the goal is to minimize the total moved distance. For delay-tolerant tasks, it performs Opportunistic Sensing and the goal is to minimize the total number of workers. These works consider the complex but actual spatial-temporal requirements of the tasks and continue to study the user selection problem in depth. However, the predicted or probabilistic trajectories used in these works can hardly be obtained and applied well in practical settings, since the fine-grained and personalized mobility prediction is so difficult in MCS and the privacy issues also need to be noticed. In our work, we simplify the mobility prediction to PoI based prediction and obtain the probabilistic utility of users to select better user sets, in which users perform the tasks in collaboration.

En Wang $et\ al.$ [15] use the similar idea of PoI based mobility prediction and propose the efficient prediction-based user recruitment strategy for mobile crowdsensing, which mainly concerns the cost of data uploading. In this work, users are divided into two groups: Pay as you go (PAYG) and Pay monthly (PAYM). A PAYG user can forward and upload the sensing data freely when he/she encounters a PAYM user. Then they formalize this user recruitment problem as recruiting the user of the highest contact probability with PAYM users and propose the prediction based strategy. This work mainly focuses on the forwarding and uploading but less on the spatio-temporal-sensitive tasks, which is quite different from our work.

2.2 Frameworks in Mobile CrowdSensing

As mentioned above, many researchers pour their time and energy into the study of user selection and propose some novel frameworks for mobile crowdsensing. Bin Guo $et\ al.$ [16] focus on the multitask-oriented worker selection framework, called *ActiveCrowd*. Lingjun Pu $et\ al.$ [5] present a QoS-oriented self-organized mobile crowdsourcing framework, called *Crowd Foraging*, where a mobile user can self-organize her task crowdsourcing in realtime. In these frameworks, they carefully consider the user recruitment process and propose their novel frameworks. However, their frameworks have strong restrictive conditions and lack consideration for the centralized and privacy issues. Some distributed and privacy-preserving frameworks [17, 18] have been proposed to solve these issues. In this paper, we carefully consider the user selection process and design the system architecture of the *Crowd UserS* framework. Then we deal with the centralized and privacy issues by using the distributed storage and computing architecture. Finally, a prototype system is implemented on the Android platform.

3 SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we introduce the spatial-temporal-sensitive mobile crowdsensing, followed by the formulation of user selection problem.

3.1 System Model

We consider a spatial-temporal-sensitive mobile crowdsensing scenario where sensing tasks have the diverse requirements of times and places. More specifically, some requesters want to get some timely sensing data, then launch many sensing tasks with different spatial-temporal requirements, denoted by $\mathcal{S} = \{s_1, s_2, ..., s_m\}$. These tasks could be aggregated into different PoIs according to their locations, and the set of PoIs can be denoted by $\mathcal{L} = \{l_1, l_2, ..., l_L\}$. Hence, each task s_i can be seen as a three tuple $< l_i, ts_i, te_i >$, in which ts_i and te_i represent the beginning time and ending time, respectively.

Note that the PoIs can be seen as some small areas, and the area size could be set at a suitable value, $e.g.$ $300m^2$, which is available for a user to perform tasks[1]. Reaching one PoI means users can perform the tasks in this area. We believe that this setting is realistic, since the target sensing locations in most MCS applications actually can be seen as some PoIs. For the area surveillance tasks, such as environment monitoring, the sensed values in one small area are almost the same, which makes it a PoI by definition. For the location aware tasks, such as the example of the queuing situation at 12:00 in Fig. 1, the setting of PoI is also realistic, since users would be glad to walk a short distance to perform the task in its PoI under the incentive. In fact, the setting of PoI is very helpful for MCS in dealing with the position error and protecting privacy partially.

[1] The size configuration is an important research problem, while it is not the main concern of this paper. The large PoIs may lead to the rough prediction and the small PoIs would increase the computation overhead. In this work, we set the size to $300m^2$, since it is the suitable distance with no overlap. We also add some experiments to test different PoI radiuses.

Then, we consider that many users move around in the scenario, denoted by $\mathcal{U} = \{u_1, u_2, ..., u_n\}$. If the user has been selected, the reward should be given to cover his cost and encourage his participation [19]. The reward is difficult to determine and many studies focus on the incentive mechanism. To simplify, we assume the users have the same reward[2], denoted by c, and it could incent the user to perform tasks. Thus, the budget constraint in this paper can be simplified to the participant number constraint, shown as $k = \lfloor \mathcal{B}/c \rfloor$. After performing the sensing task, the uploading problem also needs to be considered. We present two uploading ways, cellular links or collection points. For cellular links, it means that users upload data through cellular networks by their mobile devices. For collection points, it means that users hold the data until they reach some small areas where users can upload data for free, such as a shopping mall with free WiFi. Thus, we consider these free collection points as PoIs and users can upload the data at these collection points. Then, the H-MCS is conducted as follows.

To begin with, the task requesters need some timely sensing data and they launch the sensing task set $\mathcal{S}$. Note that $\mathcal{S}$ may change since requesters could launch tasks at any time. Each task s_i has its own attributes, $< l_i, ts_i, te_i >$. Mobile users who are willing to participate in crowdsensing have been added to the candidate set $\mathcal{U}$. They move around in the network and may arrive at the PoIs $\mathcal{L}$ at particular times t. We should notice that if and only if the mobile user u_j arrives l_i at time $t \in [ts_i, te_i]$, u_j has the ability (or willingness) to perform the task s_i. Then, the user would upload the data over the cellular link directly or hold and upload the sensing data when he arrives at the free collection point w_k. We define the probability of u_j performing with s_i and uploading the sensing data as $P(u_j, s_i)$, which can be derived from the PoI based Mobility Prediction Model. Finally, using the probabilities $\mathcal{P}$, we can obtain the expected numbers of tasks completed ($i.e.$, performed and uploaded) by the users. Then the requesters would select at most k users to collaboratively complete tasks as many as possible under a budget constraint $\mathcal{B}$ and the users' cost c. Note that the requesters should get higher revenues than what they paid, otherwise the requesters would not launch tasks and recruit the users.

3.2 Problem Formulation

After introducing the system model above, we focus on the user selection problem in H-MCS. In order to close to reality, we consider that the requesters usually launch so many tasks and some of them would not be completed. The reasons are manifold, $e.g.$ remote locations, strict time limits, special sensing equipment and so on. Therefore, the objective in H-MCS is actually to complete more tasks, which is realistic. Meanwhile, mobile crowdsensing places an emphasis on "*Crowd*", which means that there are so many users willing to participate in crowdsensing. Normally, the requesters do not have such a large budget to recruit all of the users. Hence, the question is coming: *How to select a user subset $\mu \subseteq \mathcal{U}$ to complete the most tasks under a budget constraint*? Then, the user selection problem in H-MCS can be formalized as follows:

$$maximize \quad \Sigma_{s_i \in \mathcal{S}} E(s_i, \mu) \tag{1}$$

$$s.t. \quad \mu \subseteq \mathcal{U} \tag{2}$$

$$\Sigma_{u_i \in \mu} c \leq \mathcal{B} \tag{3}$$

Here, $E(s_i, \mu)$ is the expected probability that the users in the selected set μ will complete the task s_i. Note that each task s_i has its own beginning and ending time, and it is so hard to consider all the spatial-temporal requirements in union. We could use the PoI based mobility prediction model to get the expected probability $E(s_i, \mu)$, which will be discussed at length in Section 4.

4 POI BASED MOBILITY PREDICTION

In this section, we focus on the PoI based mobility prediction in spatial-temporal-sensitive mobile crowdsensing.

4.1 Mobility Prediction Model

As mentioned above, we present a complex but practical system model, where the sensing tasks have many spatial-temporal requirements. It is difficult to select users who can satisfy these spatial-temporal requirements directly, and using the users' mobility may be a possible solution. Some researches have tried to use the predicted or probabilistic trajectories or the statistical results of workers' history records. However, these predicted or probabilistic trajectories can hardly be obtained and the statistical results cannot provide the good enough predictions in practical settings. On one hand, the fine-grained mobility prediction is difficult, especially for the large-scale MCS applications. On the other hand, it would cost a lot on personalized mobility prediction, especially for a large number of users. Additionally, the privacy issues also need to be noticed.

In light of these problems, we propose the PoI based mobility prediction model for spatial-temporal-sensitive mobile crowdsensing. The basic idea is to simplify the common prediction model from the full map to some small areas containing the locations of sensing tasks, $i.e.$, PoIs. This setting is suitable for MCS, since the target sensing locations in most MCS applications actually can be seen as some PoIs. For the area surveillance tasks, such as environment monitoring, the sensed values in one small area are almost the same, which makes it a PoI by definition. For the location aware tasks, such as the example of the queuing situation at 12:00 in Fig. 1, the setting of PoI is also realistic, since users would be glad to walk a short distance to perform the task in its PoI. In fact, the setting of PoI is very helpful for MCS in dealing with the position error and protecting privacy partially. Under this setting, we only need to make the predictions on several PoIs while ignore the useless predictions.

Note that the size and spatial distribution of the PoIs are not regular, the location based mobility prediction may be not suitable. While the PoIs can be seen as the states, then the users' movements among PoIs can be seen as the transitions between states, and the Markov Models may

[2] The setting of reward is not the point of this article, and the different rewards can also be introduced to our problem. We can add the reward as the divisor at our utility function, thus we can select the user who contribute more and cost less.

behave better. Thus, in this work, we use the Semi-Markov Process Model [14, 15, 20] and focus on the time-dependent transition probabilities between states. The associated time-dependent semi-Markov kernel $Z(\cdot)$ is defined by Eq. (4).

$$\begin{aligned} Z_u(i,j,T) =& P(S_u^{n+1}=j, t_u^{n+1}-t_u^n \le T | S_u^0, ..., S_u^n; \\ & t_u^0, ..., t_u^n) \\ =& P(S_u^{n+1}=j, t_u^{n+1}-t_u^n \le T | S_u^n = i) \end{aligned} \tag{4}$$

$Z_u(i,j,t)$ is the probability that user u will move from his current PoI i to his next PoI j at, or before time T. S_u indicates the user's sequence of PoIs and t_u is corresponding arrival times. Note that user's next PoI is associated with his current location and we can derive the probability P from the statistical results of user's history records. Then we obtain another kernel $Q(\cdot)$, denoted by Eq. (5).

$$Q_u(i,j,T) = \begin{cases} \Sigma_{l=1}^{L}\Sigma_{t=1}^{T}(Z_u(i,l,t)-Z_u(i,l,t-1))\cdot \\ Q_u(l,j,T-t), \quad i \ne j \\ 1-\Sigma_{l=1,l\ne i}^{L} Z_u(i,l,T)+ \\ \Sigma_{l=1,l\ne i}^{L}\Sigma_{t=1}^{T}(Z_u(i,l,t)-Z_u(i,l,t-1))\cdot \\ Q_u(l,i,T-t), \quad i=j \end{cases} \tag{5}$$

Note that mobile users cannot move from one PoI to another in $T=0$, so that we obtain $Q_u(i,i,0)=1$ and $Q_u(i,j,0)=0 (i \ne j)$. In fact, $Q(\cdot)$ is a recursive function and indicates the probability that user u will move from the PoI i to j just at the time T. When $i \ne j$, we consider the relay state transitions as $i \rightarrow k \rightarrow j$ and obtain the total probability. When $i=j$, we further consider the PoI sojourn probability. Finally, we get the $Q(\cdot)$ representing the probabilities that user arrive PoIs at some time. Then we can use the probabilities to calculate the expectation of users' contribution, that is, the users' utility value.

4.2 User Utility

As mentioned in problem formulation, the contribution, $i.e.$, the expected number of tasks completed by the selected users, can be seen as the utility value, denoted by $\Sigma_{s_i\in\mathcal{S}}E(s_i,\mu)$. In it, $E(s_i,\mu)$ denotes the expected probability that all the users in μ collaboratively complete the task s_i. We consider the users in the selected set μ are independent. They would perform the same task s_i in collaboration. In this case, the probability of completing the task is actually a joint probability, known as the $joint\ completing\ probability$. Then, $E(s_i,\mu)$ can be calculated as Eq. (6).

$$E(s_i,\mu) = 1-\Pi_{u_j\in\mu}(1-P(u_j,s_i)) \tag{6}$$

Eq. (6) shows that the task will be completed as long as one of the selected users completes it[3]. $P(u_j,s_i)$ indicates the probability of user u_j completing task s_i as mentioned above. The temporal and spatial requirements will be satisfied by $P(u_j,s_i)$. It can be calculated by the $Q(\cdot)$ in the PoI based mobility prediction model.

Different from previous studies, we consider that completing a task means not only performing but also uploading the sensing data. We believe that the uploading process is necessary and practical to be considered. As mentioned above, we present two uploading ways: cellular links and collection points for two common kinds of tasks, time-sensitive tasks and delay-tolerant tasks [16]. For time-sensitive tasks, users need to upload the sensing data immediately after performing sensing tasks, thus they should use the cellular links. For delay-tolerant tasks, users can hold and upload the sensing data before the deadline, thus they would like to use free collection points in order to reduce their costs.

From the perspective of mobility, the main difference between the two uploading ways is that users need to move to one free collection point after performing tasks. Note that we do not propose to use ad-hoc connections to transfer the sensed data to the free collection point, since this method may cost a lot on the transmissions and the connecting predictions may cause privacy issues. Thus, two uploading ways should have different Ps. In other words, where and how to upload the sensing data has a great influence on $P(u_j,s_i)$. Users may upload data directly over the cellular infrastructure or hold and upload it when they reach the free collection points after performing the sensing tasks. If the user u_j decides to upload data directly over cellular links, we just need to consider the probability of arriving at the destination PoIs in time. If the user chooses to use the collection points, we should further consider the probability of arriving at one of the free collection points after performing task s_i. Fortunately, all the results are probabilistic and we could obtain a unified representation, $P(u_j,s_i)$. We will discuss them as follows.

4.2.1 *Mobile CrowdSensing over the cellular infrastructure*

Most of existing works on mobile crowdsensing assume that mobile users upload their sensing data from PoIs through the cellular link immediately. In this case, we only take the "perform" into account while not being distracted by the uploading. Then, $P(u_j,s_i)$ can be calculated by Eq. (7).

$$P(u_j,s_i) = 1-\Pi_{t=ts_i}^{te_i}(1-Q_{u_j}(l_j,l_i,t)) \tag{7}$$

$Q_{u_j}(l_j,l_i,t)$ is the probability that user u_j moves from its current location l_j to task location l_i at time t. It can be obtained by the mobility prediction model. We calculate $P(u_j,s_i)$ by using $Q_{u_j}(l_j,l_i,t)$ and consider the temporal requirements as variables. Then, the different tasks with different requirements can be considered in union. Here, when user u_j arrives at the task PoI l_i during the task's lifetime $[ts_i,te_i]$, he could perform the task. After performing the task successfully, user u_j will upload the sensing data to requesters directly through the cellular link, such as 4G.

4.2.2 *Mobile CrowdSensing over the collection points*

Most tasks in mobile crowdsensing need a large number of sensing data. Obviously, it costs too much when the users upload data through the cellular link directly. Many tasks

[3] For the tasks which need sensor readings from a larger number of users, our 'joint completing probability' should be modified from 'at least 1 user' to 'at least k users', shown as $1-\Pi_{u_j\in\mu}(1-P(u_j)) - \sum_{\hat{\mu}\in\mu,|\hat{\mu}|=k}\Pi_{u_k\in\hat{\mu}}P(u_k)\cdot\Pi_{u_j\in\mu\setminus\hat{\mu}}(1-P(u_j))$. Note that the computation overhead will rapidly increase along with the growth of number of users, which may be not suitable in MCS.

have real-time requirements as we discussed above. Hence, many solutions have been introduced to reduce the costs and meet the real-time requirements [10, 21]. Among them, Merkouris Karaliopoulos $et\ al.$ [4] propose that users could hold and upload the sensing data when they reach free collection points, such as $WiFi$ and $RoadsideAPs$. In this case, users will perform the sensing tasks successfully first. Then, they should hold and upload the data later when they reach the collection points. As introduced above, we represent the set of collection points by using $\mathcal{W} = \{w_1, w_2, ..., w_W\}$. Here, we should further consider that users with sensing data need to move to one of the collection points, w_k. As shown in Eq. (6) and Eq. (7), users arriving at PoIs at different times can be seen as independent events. Hence, we define the probability that users will move to collection points in time as the following equation:

$$R(u_j, s_i, t_{ji}) = 1 - \Pi_{t=t_{ji}}^{te_i}(1 - C(u_j, l_i, t - t_{ji})) \tag{8}$$

Eq. (8) shows that u_j moves from the destination PoI of task s_i to the collection points before the task's deadline, te_i. Note that t_{ji} denotes the time when u_j performs task s_i, and the expression $t = t_{ji}$ means u_j moves to collection points from the time t_{ji}. Specifically, we consider that the PoI l_i and a collection point w_k may have the same location. In this case, it is equivalent for users to upload data through cellular infrastructures. As a result, we use the equal sign here.

Meanwhile, $C(u_j, l_i, t)$ is the probability that user u_j moves from task location l_i to any of the collection points. Assuming that there is no overlap between the collection points, it can be seen as an exclusion event that users arrive at different collection points at the same time. Then, we obtain $C(u_j, l_i, t)$ by using $Q(\cdot)$ as follows:

$$C(u_j, l_i, t) = \sum_{w_k \in \mathcal{W}} Q_{u_j}(l_j, w_k, t) \tag{9}$$

Note that we use l_i as the initial state of user u_j. If l_i and w_k may have the same location, we can obtain that $Q_{u_j}(l_i, l_i, 0) = 1$. That is to say that user u_j can upload his sensing data directly through the collection points at the same location.

Then, $P(u_j, s_i)$, the probability of performing the task and uploading the data can be rewritten by using Eq. (10).

$$P(u_j, s_i) = 1 - \Pi_{t=ts_i}^{te_i}(1 - Q_{u_j}(l_j, l_i, t) \cdot R(u_j, s_i, t)) \tag{10}$$

Here, when user u_j arrives at the task PoI l_i during the task's lifetime $[ts_i, te_i]$, he could perform the task. After performing the task successfully, user u_j will move to any collection point and upload the sensing data to requesters through the free collection points, such as WiFi and Roadside APs.

In summary, we could calculate $P(u_j, s_i)$, the probability that user u_j will complete the task s_i, through whether we upload data over cellular links or collection points. Using the probabilistic expression $P(u_j, s_i)$, we can get the expected number of tasks completed by a selected user set, which is defined as $\Sigma_{s_i \in \mathcal{S}} E(s_i, \mu)$, *i.e.*, the users' utility function. To simplify the notation, we define our objective function, $\Sigma_{s_i \in \mathcal{S}} E(s_i, \mu) = f(\mu)$. We use the $Q(\cdot)$, obtained from PoI based mobility prediction model, to deal with the various temporal and spatial requirements of tasks in union by using $P(u_j, s_i)$. Based on that, we propose a greedy offline user selection algorithm to select a user set that will complete as many tasks as possible. Furthermore, we extend the scenario to a more general one and propose an online algorithm. We will discuss the two algorithms at the next section.

Algorithm 1 The g-MUS Algorithm

```
Input: $\mathcal{S}$: a set of tasks, $\mathcal{U}$: a set of users with their $Q$, $k$: the maximum number of selected users, $k = \lfloor \mathcal{B}/c \rfloor$, $\phi$: the available user set
Output: $\mu$: the selected user set
Calculate $Q$ for each use
Initialize $\phi = \mathcal{U}$ and $\mu = \emptyset$
while $|\mu| < k$ do
    Select one user $u_i$ from $\phi$ to maximize $\theta_{u_i} = f(\mu \cup \{u_i\}) - f(\mu)$
    Update $\mu = \mu \cup \{u_i\}$ and $\phi = \phi \setminus \{u_i\}$
return $\mu$.
```

5 Offline User Selection

In this section, we analyze the hardness of user selection problem in spatial-temporal-sensitive mobile crowdsensing, and then present a greedy offline algorithm.

5.1 Problem Hardness

Before introducing the proposed greedy algorithm, we first prove the NP-hardness of the user selection problem in H-MCS, as shown in the following theorem.

Theorem 1. *The prediction based user selection problem in H-MCS is NP-hard.*

Proof. The foundation of user selection problem in our paper is PoI based mobility prediction, for which we could consider a special case, that is to give the predetermined mobility information. Here, we could get $Q_{u_j}(l_j, l_i, t) \in \{0, 1\}$, which means that user u_j would reach l_i at t or not. Similarly, $P(u_j, s_i) \in \{0, 1\}$ means u_j would complete the task s_i or not. Then, we can get the task set "covered" by user u_j, denoted as S_j. If the tasks have the same cost, we could select k users under the participant number constraint. Then, the problem is indeed a classic NP problem, *Max k-cover* [22]: given a collection of task set $\{S_1, S_2, ..., S_n\}$, each task set will cover several tasks $S_j = s_{j1}, s_{j2}, ...$, then the objective is to select k sub-collections of $\{S_1, S_2, ..., S_n\}$ to cover the most tasks. That is to say, the special case is NP-hard. Consequently, the prediction based user selection problem is also at least NP-hard. The theorem holds. □

5.2 Greedy Algorithm

The prediction based user selection problem in H-MCS is NP-hard and there are so many sub-collections which conform to the participant number constraint. In other words, the user selection problem has such a large solution space that performing an exhaustive search is not feasible. Hence, we use the greedy heuristic strategy to approximately solve the problem.

Now we are ready to describe our greedy user selection algorithm based on mobility prediction, called g-MUS, as shown in Algorithm 1. First of all, we should get the $Q(\cdot)$ for each user and use it to predict the user's PoI based mobility (line 1). We can obtain the result by processing the history spatial-temporal traces according to Eq. (4) and Eq. (5). We initialize the available user set and selected user set (line 2). Our algorithm will select one of the available users in each iteration of the while-loop (line 3 to 5). The selected user in each round should have the maximum gain $\theta_{u_i} = f(\mu \cup \{u_i\}) - f(\mu) = \Sigma_{s_i \in \mathcal{S}} E(s_i, \mu \cup \{u_i\}) - \Sigma_{s_i \in \mathcal{S}} E(s_i, \mu)$, in which $E(\cdot)$ is calculated based on Eq. (6) and Q). The maximum gain ensures that we select the one who will contribute the most in collaboration with the selected users (line 4). After one user has been selected, we update the selected set and available set, respectively. Finally, we obtain the selected user set.

5.3 Performance Analysis

Mobility prediction is the basis for our user selection algorithms, in other words, the performance analysis depends on the ideal assumption of perfect mobility prediction. To simplify the notation, we use $f(\mu)$ to replace $\Sigma_{s_i \in \mathcal{S}} E(s_i, \mu)$, as defined above. We first prove the property of the objective function.

Theorem 2. *1)$f(\emptyset) = 0$; 2)$f(\mu)$ is increasing and submodular.*

Proof. 1) $\mu = \emptyset$ means that no user has been selected and no one would perform tasks. Then, $E(s_i, \emptyset) = 0$ for each $s_i \in \mathcal{S}$, according to Eq. (6). Thus, $f(\emptyset) = \Sigma_{s_i \in \mathcal{S}} E(s_i, \emptyset) = 0$.
2) We first prove that $f(\mu)$ is an increasing function. Without loss of generality, we have two user subsets, μ_1 and μ_2 and $\mu_1 \subseteq \mu_2$. Then, we obtain the Eq. (11) for each $s_i \in \mathcal{S}$. Here, we define that $\mu_3 = \mu_2 \setminus \mu_1$ and $P(u_j, s_i)$ can be calculated by Eq. (7) and Eq. (10) according to their uploading ways. While $P(u_j, s_i)$ represents the probability of uj performing si and uploading the sensing data, we can get that $0 \leq P(u_j, s_i) \leq 1$ ($\forall\ u_j \in \mathcal{U}, s_i \in \mathcal{S}$), according to Eq. (7) and Eq. (10). Then, $E(s_i, \mu_1) - E(s_i, \mu_2) \leq 0$ for each $s_i \in \mathcal{S}$ and $\Sigma_{s_i \in \mathcal{S}} E(s_i, \mu_1) - \Sigma_{s_i \in \mathcal{S}} E(s_i, \mu_2) \leq 0$. Therefore, $f(\mu)$ is an increasing function.

$$\begin{aligned} & E(s_i, \mu_1) - E(s_i, \mu_2) \\ =& (1 - \Pi_{u_j \in \mu_1}(1 - P(u_j, s_i))) - (1 - \Pi_{u_j \in \mu_2}(1 - P(u_j, s_i))) \\ =& \Pi_{u_j \in \mu_2}(1 - P(u_j, s_i)) - \Pi_{u_j \in \mu_1}(1 - P(u_j, s_i)) \\ =& \Pi_{u_j \in \mu_1}(1 - P(u_j, s_i)) \cdot (\Pi_{u_j \in \mu_3}(1 - P(u_j, s_i)) - 1) \leq 0 \end{aligned} \tag{11}$$

Similarly, we prove the $f(\mu)$ is submodular. Here, we define $u_i \in \mathcal{U} \setminus \mu_2$, and obtain Eq. (12).

$$\begin{aligned} & (f(\mu_1 \cup \{u_i\}) - f(\mu_1)) - (f(\mu_2 \cup \{u_i\}) - f(\mu_2)) \\ =& (\Sigma_{s_i \in \mathcal{S}} E(s_i, \mu_1 \cup \{u_i\}) - \Sigma_{s_i \in \mathcal{S}} E(s_i, \mu_1)) - \\ & (\Sigma_{s_i \in \mathcal{S}} E(s_i, \mu_2 \cup \{u_i\}) - \Sigma_{s_i \in \mathcal{S}} E(s_i, \mu_2)) \\ =& \Sigma_{s_i \in \mathcal{S}} (\Pi_{u_j \in \mu_1}(1 - P(u_j, s_i)) \cdot P(u_i, s_i) - \\ & \Pi_{u_j \in \mu_2}(1 - P(u_j, s_i)) \cdot P(u_i, s_i)) \\ =& \Sigma_{s_i \in \mathcal{S}} \Pi_{u_j \in \mu_1}(1 - P(u_j, s_i)) \cdot P(u_i, s_i) \cdot \\ & (1 - \Pi_{u_j \in \mu_3}(1 - P(u_j, s_i))) \geq 0 \end{aligned} \tag{12}$$

As discussed above, we know that $0 \leq P(u_j, s_i) \leq 1$ ($\forall\ u_j \in \mathcal{U}, s_i \in \mathcal{S}$). Then, we obtain that $f(\mu_1 \cup \{u_i\} - f(\mu_1) \geq f(\mu_2 \cup \{u_i\}) - f(\mu_2)$ according Eq. (12). Therefore, the submodular property of $f(\mu)$ holds. □

Now, we can give the approximation ratio of the greedy strategy by the following theorem.

Theorem 3. *The proposed greedy strategy can achieve a $(1 - \frac{f(u_{max})}{f(\mu^*) - |\mu|})$-approximation solution, where u_{max} is the best user in the first round of the greedy strategy and μ^* is the optimal user set.*

Proof. Let $u_1, u_2, ..., u_k$ be the sequence of users selected by the greedy strategy, u_{k+1} is the next one if we have a larger budget, and their costs are $c_1, c_2, ..., c_k, c_{k+1}$. Set $\mu_0 = \emptyset$ and $\mu_i = u_j : 1 \leq j \leq i$. μ_k is the selected subset under the budget B. Then, we obtain that for each $1 \leq i \leq k$,

$$\begin{aligned} & \frac{f(\mu_{i-1} \cup \{u_i\}) - f(\mu_{i-1})}{c_i} \\ \geq & max_{u_* \in \mu^*} \frac{f(\mu_{i-1} \cup \{u_*\}) - f(\mu_{i-1})}{c_*} \\ \geq & \frac{\sum_{u_* \in \mu^*} (f(\mu_{i-1} \cup \{u_*\}) - f(\mu_{i-1}))}{\sum_{u_* \in \mu^*} c_*} \\ \geq & \frac{f(\mu^*) - f(\mu_{i-1})}{B} \end{aligned} \tag{13}$$

Note that the optimal solution also has the same budget constraint, so we have $\sum_{u_* \in \mu^*} c_* \leq B$. Also, the function $f(\mu)$ is submodular according to Theorem 2. Thus, the last inequality in Eq. (13) holds. Then, we obtain the cost of u_i and the total costs of μ,

$$c_i \leq \frac{B}{f(\mu^*) - f(\mu_{i-1})} \cdot (f(\mu_{i-1} \cup \{u_i\}) - f(\mu_{i-1})) \tag{14}$$

Then, we could get the total costs of the selected subset μ. We add all the costs in μ and the extra c_{k+1} together to get the lower bound B,

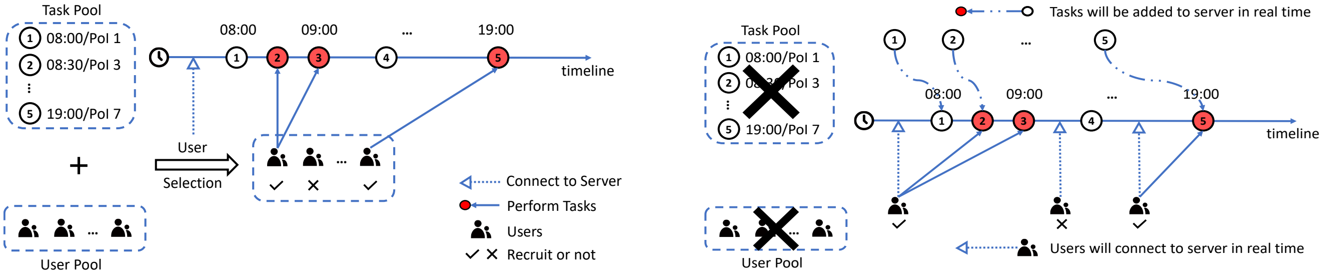
$$\begin{aligned} B < & \sum_{u_i \in \mu} c_i + c_{k+1} \\ \leq & \frac{B}{f(\mu^*) - f(\mu_{k-1})} \cdot (f(\mu_{k-1} \cup \{u_k\}) - f(\mu_{k-1})) + ...+ \\ & \frac{B}{f(\mu^*)} \cdot f(u_{max}) + \\ & \frac{B}{f(\mu^*) - f(\mu_k)} \cdot (f(\mu_k \cup \{u_{k+1}\}) - f(\mu_k)) \end{aligned} \tag{15}$$

Here, μ_k is the selected subset under the budget B. If we add the extra user u_{k+1}, the total costs should be out of the budget, then the Eq. (15) holds. Note that u_{max} have the biggest $f(u)$, so we obtain the Eq. (16).

$$B < \frac{B}{f(\mu^*) - f(\mu_k)} \cdot f(u_{max}) + |\mu| \cdot \frac{B}{f(\mu^*)} \cdot f(u_{max}) \tag{16}$$

The objective is to get the relationship between $f(\mu_k)$ and $f(\mu^*)$, so we simplify the Eq. (16). As mentioned in the framework model, requesters should get higher revenues than what they paid, which means $\frac{f(\mu^*)}{|\mu|} > 1$. So, we have $f(\mu^*) - |\mu| > 0$ and obtain the Eq. (17).

$$f(\mu_k) > (1 - \frac{f(u_{max})}{f(\mu^*) - |\mu|}) \cdot f(\mu^*) \tag{17}$$



(a) offline scenario (b) online scenario

Fig. 2: Illustration of offline/online user selection: (a)offline: users and tasks are predetermined before the start of MCS; (b)online: tasks and users would be coming in real time and we must decide whether to select the user or not immediately.

Thus, the approximation ratio is $1 - \frac{f(u_{max})}{f(\mu^*)-|\mu|}$, according to the Eq. (17). The theorem holds. □

In brief, the greedy criterion of our algorithm is to select the user who contributes the most with the least cost first. The complexity of the user selection algorithm consists of two parts, the PoI based mobility prediction model and the iterative selection process. For mobility prediction, we get the size per user of the Q matrix (L^2T), where L is the number of PoIs and T donotes the prediction window. As shown in Eq. (5), calculating $Q(\cdot)$ is actually an iterative process. These results will be very sparse in the real world [14] and could be done before the user selection. For the iterative selection process, the computation overhead is dominated by Line 4, and the worst case is $O(n^2mT)$. Here, T is the lifetime of tasks and is relevant to the prediction window.

6 Online User Selection

In this section, we extend our problem to a more practical scenario, called online scenario, where tasks and users would be coming in real time and we must decide whether to select the user within a short time.

6.1 Online Scenario

In our initial problem, we consider that all the users and tasks are predetermined before the start of MCS, and no further tasks and users can come after MCS starts. Then, the server uses the offline user selection algorithm to select some users to perform these predetermined tasks, as shown in Fig. 2 (a). The offline algorithm works well when the mobile crowdsensing is scheduled before the beginning. However, in real world, the tasks and users could arrive at any time. Moreover, the users would like to know whether to be recruited or not within a short time after they are coming. Hence, we extend our problem to this practical online scenario where tasks and users would be coming in real time and we must decide whether to select the user or not immediately, as shown in Fig. 2 (b).

Note that all the users should record their traces before participating into MCS, otherwise we cannot do the mobility predictions and the user selection will make no sense. Also, users would not participate in MCS all the time. We define the working time of user as his activetime, and the length is much less than the total time. We consider that the users working at different activetimes would cover different sensing tasks. Thus, we ignore the coming times but focus on how many tasks could be ”covered” by users, that is, the contributions obtained by Eq. (6) and Q. Then, we can use the results to determine whether to select or wait for a better one.

Algorithm 2 The o-MUS Algorithm

```
Input: $\mathcal{S}$: a set of tasks, $\mathcal{U} = u_1, u_2, ..., u_n$: a stream of users with their $Q$, sorted by their coming time, $k$: the maximum number of selected users, $k = \lfloor \mathcal{B}/c \rfloor$, $l$: the length of a phase, $l = \lceil n/k \rceil$, $\varepsilon$: the utility threshold
Output: $\mu$: the selected user set
1:  Calculate $Q$ for each user.
2:  Initialize $\phi = \mathcal{U}$, $\mu = \emptyset$, and $i = 0$.
3:  while $i < n$ do
4:      $i++$                                   ▷ $u_i$ is coming
5:      if $k - |\mu| \geq n - i$ then
6:          $\mu = \mu \cup \{u_i\}$               ▷ recruit all the rest
7:      else
8:          if $i\%l == 1$ then
9:              $\varepsilon = 0$          ▷ initialize the threshold at a new segment
10:         if $i\%l \leq \lfloor l/e \rfloor$ then                ▷ observe
11:             Calculate $\theta_{u_i} = f(\mu \cup \{u_i\}) - f(\mu)$
12:             Update $\varepsilon = max\{\varepsilon, \theta_{u_i}\}$
13:         else if $i\%l > \lfloor l/e \rfloor$ then            ▷ select
14:             Calculate $\theta_{u_i} = f(\mu \cup \{u_i\}) - f(\mu)$
15:             if $\theta_{u_i} \geq \varepsilon$ then
16:                 Update $\mu = \mu \cup \{u_i\}$
17:                 $i = l \cdot |\mu| + 1$        ▷ ignore the rest in this segment
18:                 Continue
    return $\mu$
```

6.2 Online Algorithm

The user selection problem in online scenario is much more challenging. When a user connects to the server, we have to decide whether to recruit or not immediately, without the knowledge of future users. Moreover, the users perform the sensing tasks collaboratively, which means that the selected users would influence the next selection (*i.e.*, the submodular objective function $f(\mu)$), and it makes the problem more complex. Fortunately, the user selection problem is very suitable to formulate as a variant of famous secretary problem, *submodular secretary problem*[23].

The basic form of the secretary problem is to hire the best secretary out of n rankable applicants. The applicants would be coming one by one, and decisions are made immediately. Once rejected, an applicant cannot be recalled. Further considering the multiple secretaries and the submodular utility function, MohammadHossein Bateni *et al.* proposed the *submodular secretary problem*, in order to select k secretaries so as to maximize the expectation of a submodular function which defines efficiency of the selected secretarial group based on their overlapping skills.

In our user selection problem, we consider that a total of n users will connect to server during the MCS campaign. Users would be coming in real time and the decisions must be made immediately after the connection. The goal can actually be interpreted as to select $k = \lfloor \mathcal{B}/c \rfloor$ users to maximize the expected number of completed tasks, which is the same submodular function in offline case, *i.e.*, $f(\mu)$. Then we use the online algorithm for *submodular secretary problem* to approximately solve the problem.

The online user selection algorithm is summarized in Algorithm 2. We partition the n users into k equally-sized segments, and try to select the best one in each segment[4]. Specifically, we should predict the users' PoI based mobility first(line 1), which is the foundation. Then, we divide the n users into k segments with the length $l = \lceil n/k \rceil$, and wait for the next user's coming. If the coming user is the first one in the segment, we initialize the utility threshold $\varepsilon = 0$ (line 9). In each segment, we observe the first $\lfloor l/e \rfloor$ users and update the threshold as the max $\theta_{u_i} = f(\mu \cup \{u_i\}) - f(\mu)$, then select the next one who has a larger θ_{u_i} (line 10-18). Finally, we obtain the selected user set.

6.3 Performance Analysis

The online user selection algorithm actually has the same goal as the offline case, while adding an ”online” condition on it. Thus, we have the same objective function $f(\mu)$ in online case, which has been proven as a monotonic increasing submodular function. Let $OPT = \{u_{i_1}, u_{i_2}, ..., u_{i_k}\}$ be the optimal solution obtained. Note that the set $i_1, i_2, ...i_k$ is a uniformly random subset of $1, 2, ..., n$, and the permutation of the selected users is also uniformly random. It is reasonable since users coming at different time will only work for a period of time (usually a short time in reality), thus they would ”cover” (perform) different tasks, which can be seen as the contributions uniformly according to our objective function. Under the monotone submodular function and uniformly random users, our proposed online user selection algorithm can be proved to achieve an expected approximation ratio of at least $\frac{1-1/e}{7}$ [23].

7 Crowd UserS Framework

In this section, we propose *Crowd UserS*, a prediction-based user selection frameworkand implement a prototype system on the Android platform.

7.1 Framework Architecture

The proposed *Crowd UserS* framework follows the client-server architecture as illustrated in Fig. 3. Note that the clients have been divided into requesters and users, while the servers are data centers or clouds.

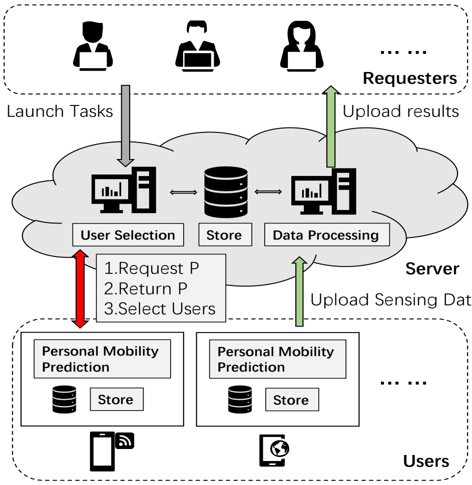


Fig. 3: The system architecture of *Crowd UserS*.

On the requesters side, the requesters could launch tasks on the server and get the results they need. We distinguish them from the server since we would like to build the *Crowd UserS* framework as a a general framework for mobile crowdsensing.

On the server side, the *Crowd UserS* would show the tasks launched by requesters. Then, the user selection algorithm should be used to select a better subset of users to complete the tasks under budget constraints.After the selected users upload their sensing data, the necessary data processing would be performed on the server. Finally, the server would return the results to requesters. In this paper, we focus on the user selection in mobile crowdsensing. Note that we have considered the different uploading ways and offline/online settings, and provide a unified representation. *Crowd UserS* has the flexible transitions on them according to the user profile or the requirements extracted from sensing tasks and scenarios.

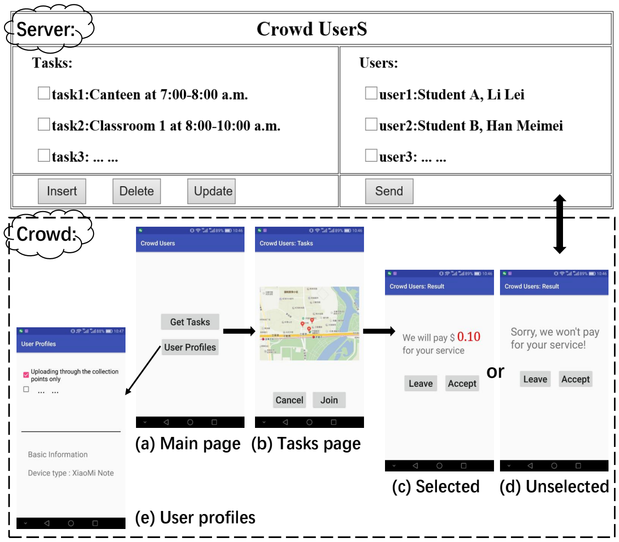
On the users side, each client is a mobile smart device carried by a user. When the user is selected, he could perform the sensing tasks using his mobile smart device, then upload the sensing data. Note that each user could record his personal mobility information and calculate the probability P locally, which is designed to reduce the centralized computing and protect privacy partially. Specifically, the server doesn't need to know the sensitive information (*i.e.*, mobility trace), which would protect privacy up to a certain point. It just requests P from users. And users can calculate it on their mobile devices by using the $Q(\cdot)$ function, which is also stored locally. The mobile devices have sufficient storage and computing capabilities, since $Q(\cdot)$ will be very sparse in the real world [14].

7.2 Prototype Implementation

We implement a prototype system of *Crowd UserS* with three components: a Web portal (requesters), an application on the Android platform (users), and a central server (server). Note that we implement the application on several off-the-shelf smartphones using Android OS 5.0+, such as Xiaomi Note and OPPO R9s, according to the framework model shown

[4] We virtually insert dummy users if n is not divisible by k. Note that we can not obtain the exact n in real world. It is acceptable that we estimate or predict n according to the historical data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2879098, IEEE Transactions on Mobile Computing



Crowd UserS

Tasks:

□task1:Canteen at 7:00-8:00 a.m.

□task2:Classroom 1 at 8:00-10:00 a.m.

□task3: ... ...

Users:

□user1:Student A, Li Lei

□user2:Student B, Han Meimei

□user3: ... ...

☑ Uploading through the collection points only

□ ... ...

Basic Information

Device type : XiaoMi Note

We will pay $ 0.10 for your service

Sorry, we won't pay for your service!

(a) Main page (b) Tasks page (c) Selected (d) Unselected (e) User profiles

Fig. 4: The system prototype of the *Crowd UserS* framework.

in Fig. 3. The main interfaces of *Crowd UserS* are shown in Fig. 4.

On the server side, the requesters could launch and modify the sensing tasks by using the *Insert*, *Delete* and *Update* functions. The server could push the tasks to users and request the user's probability P. On the crowd side, each user has his main page and could modify his user profiles, such as choosing *"Uploading through the collection points only"* to let the user specify how to upload the sensing data. When the server sends the tasks to the user and requests P, the user could get the tasks' information. If the user decides to join, his probability P would return to the server or return 0. Then, the offline greedy algorithm or the online algorithm would begin to work and provide rewards for the selected users according to the demand.

8 PERFORMANCE EVALUATION

In this section, we conduct extensive simulations on three real-life mobile traces to evaluate the performance of our proposed *Crowd UserS*.

8.1 Algorithms in Comparison

For the offline case, the user selection problem focuses on the various spatial-temporal requirements. It is quite a bit different from the existing works, and as a result, the previous algorithm cannot be used to solve our problem directly. In this paper, we design the **MAXP** algorithm modified from the related algorithms in [4], [6], and [11]. In this algorithm, the user mobility has been seen as a statistical result of a user's trace to measure the probability that he will pass by the PoI of a task. Then, the user with the maximum effective increments will be added to the selected user set μ in each round. We also implement the random selection algorithm **RAND**, and the ideal selection algorithm **DUS** with predetermined user mobility [4]. Besides, we use the suffix **-D** to represent that users upload the sensing data directly and the suffix **-C** means that users should hold and upload data at some collection points.

For the online case, it is actually an extention of the offline case on the coming times and prompt decisions. The online algorithm has the same objective and measurement of utility with the offline algorithm. Actually, the online algorithm can be seen as an approximation of the offline algorithm. We have evaluated the effectiveness of the user selection algorithm based on mobility prediction over the different uploading ways in the offline case. Thus, in the online case, we mainly focus on the comparison between the online and offline algorithms, $i.e.$, **g-MUS** and **o-MUS**, where the uploading ways are decided by the tasks randomly. In addition, we also implement the random selection algorithm **RAND** as a supplement.

8.2 Data Sets and Simulation Configurations

We conduct extensive simulations on three real-life mobile traces: *Feeder* [24], *Shanghai Traces*, and *GeoLife* [25, 26]. We will introduce these three traces respectively as follows.

The ***Feeder*** dataset contains Taxicab GPS data collected in Shenzhen, China. It provides better mobility and we select 196 taxis as users because their records are continuous and have similar periods of time. The 13 most frequently accessed Points (covered by more than 400 times) are selected as the PoIs, as shown in Fig. 5.

The ***Shanghai Traces*** dataset was collected by the taxis in Shanghai over several months. The dataset has more than 14700 records and each record represents a taxi trace. We select part of the records (310 traces) to filter some abnormal traces and 18 PoIs have also been outlined in red on Fig. 6.

The ***GeoLife*** trajectory dataset was collected by 182 users with a broad range of users outdoor movements. It contains more than $17,000$ trajectories and has a total duration of $50,000+$ hours, recorded by GPS loggers and phones. We select 727 traces with continuous and similar periods of time as users, then the thermodynamic map and the selected 13 PoIs are shown in Fig. 7.

The *Feeder* and *Shanghai Traces* datasets were collected by taxis with strong randomness and The *GeoLife* dataset with fine-grained trajectories was collected by mobile phones carried by users with a larger scale. These three widely-used datasets can measure the effectiveness of our proposed algorithms well. The tasks will be generated with different locations (PoIs) and times. Then, we compare the different algorithms by using the average number of completed tasks. The budget constraints (participant number constraints), the number of tasks, and the lifetimes of tasks would be considered in the experiments. For the online algorithm, we evaluate the activetime of users particularly.

8.3 Performances

8.3.1 Offline Algorithm

We evaluate the performances on the *Feeder*, *Shanghai Traces*, and *GeoLife* in offline case first, as shown in Fig. 8. We change the lifetime, budget, number of tasks, and number of collection points, while keeping the others fixed. The results on three datasets have the similar tendencies and show the effectiveness of our proposed prediction based user selection algorithm.

Specifically, our proposed g-MUS algorithm in *Crowd UserS* achieves a good performance. It performs better than the MAXP and RAND algorithm in most of the time. Moreover, compared with the DUS algorithm with the known

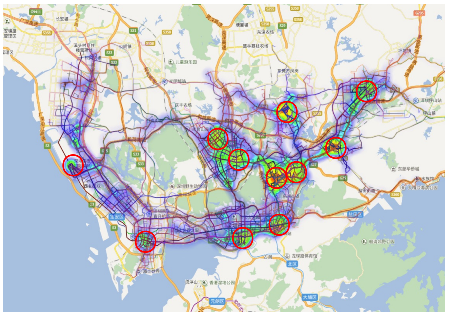


Fig. 5: PoIs selection based on the Feeder dataset.

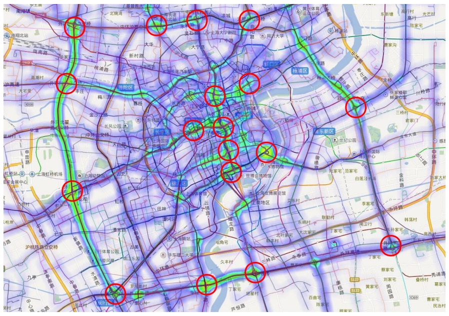


Fig. 6: PoIs selection based on the Shanghai Traces dataset.

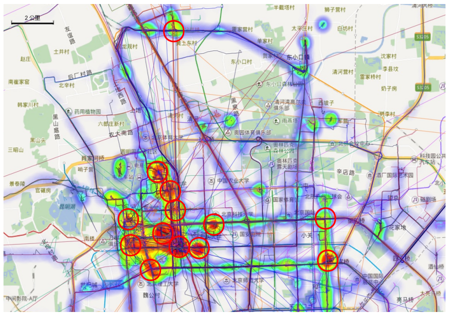
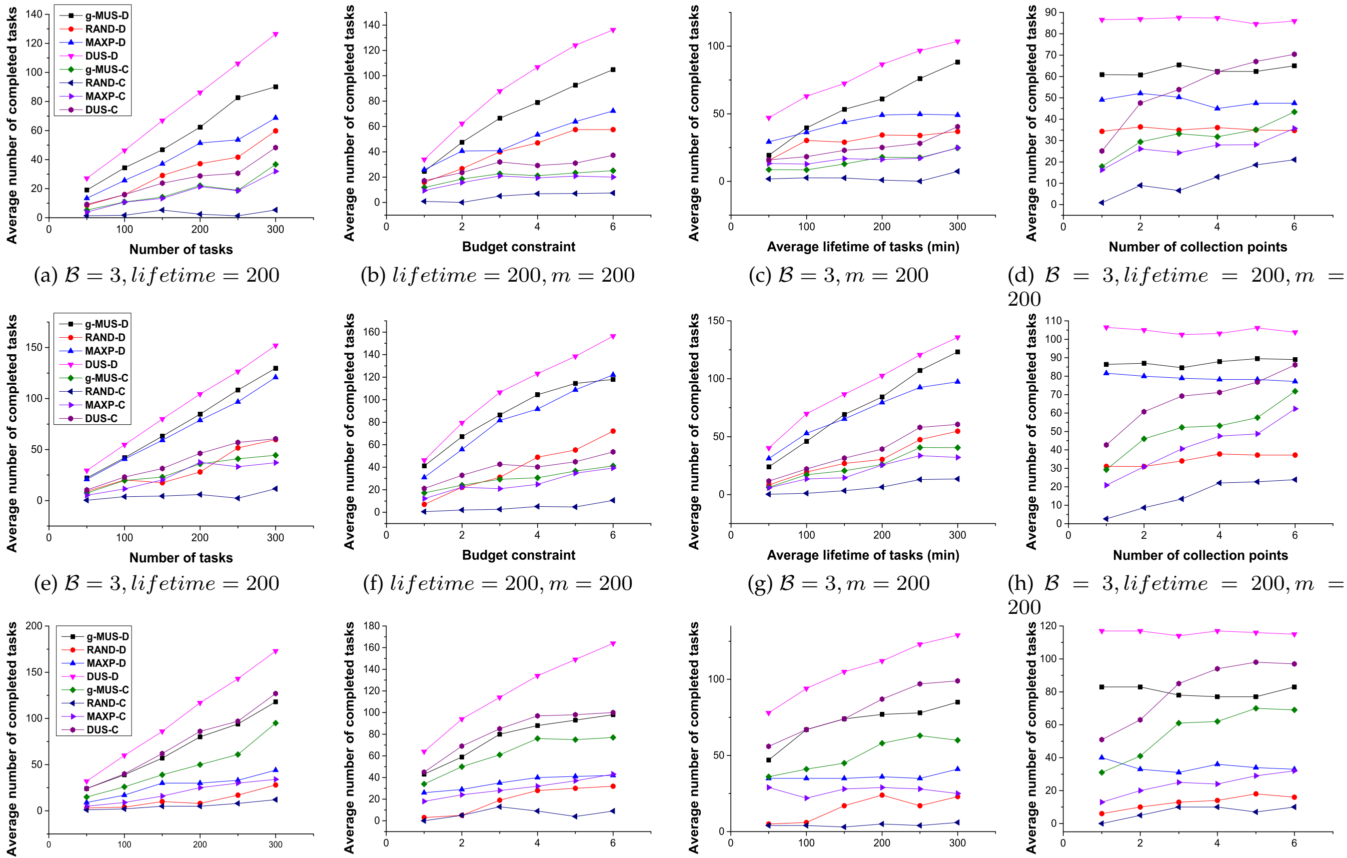


Fig. 7: PoIs selection based on the GeoLife dataset.



(a) $\mathcal{B}=3, lifetime=200$

(b) $lifetime=200, m=200$

(c) $\mathcal{B}=3, m=200$

(d) $\mathcal{B}=3, lifetime=200, m=200$

(e) $\mathcal{B}=3, lifetime=200$

(f) $lifetime=200, m=200$

(g) $\mathcal{B}=3, m=200$

(h) $\mathcal{B}=3, lifetime=200, m=200$

(i) $\mathcal{B}=3, lifetime=200$

(j) $lifetime=200, m=200$

(k) $\mathcal{B}=3, m=200$

(l) $\mathcal{B}=3, lifetime=200, m=200$

Fig. 8: The simulation results of the *Feeder*, *Shanghai Trace*, and *GeoLife* in offline case.

mobility, the users selected by our algorithm can only complete 10% to 20% fewer tasks, which also shows that the PoI based mobility prediction method achieves a good prediction accuracy. Note that the algorithms with the suffix -D have the similar trends but relatively poor performances than the ones with -C. The reason is that the users should arrive at a PoI and collection points sequentially in most cases, unless there is one of collection point in the PoI. We also change the number of collection points in Fig. 8 (d, h, and l). Along with the increase of collection points, the algorithms with suffix -C perform better and the performances nearly reach the ones with suffix -D. When there is a free collection point for every PoI, the algorithms with suffix -C and -D are actually the same.

In addition, we conduct simulations to evaluate the approximation ratio of the greedy strategy. We perform the exhaustive search to get the optimal solution, called OPT, as the comparison algorithm. The results are shown in Table 1, where our greedy strategy can achieve the inferred approximation ratio and matches the result of the theoretical analysis.

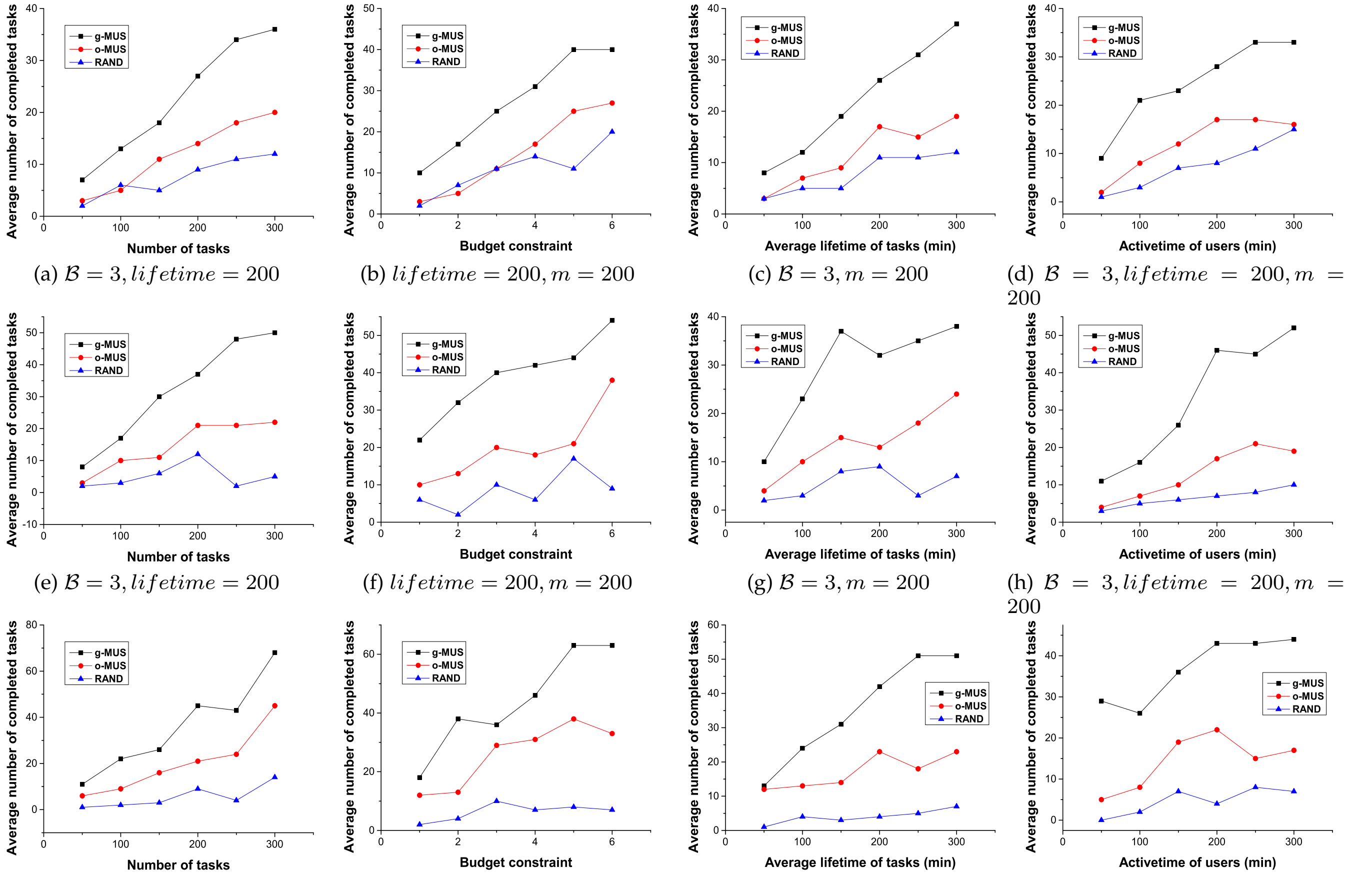
8.3.2 Online Algorithm

Our online algorithm is proposed to deal with the online problem extended from the offline setting. Actually, the online algorithm is an approximation of the offline algorithm, but it can achieve an acceptable performance for the online scenario. We evaluate the performances of three algorithms (g-MUS, o-MUS, and RAND) on the *Feeder*, *Shanghai Traces*, and *GeoLife* in online scenario.

We conduct these algorithms by changing the lifetime, budget, number of tasks, and activetime of users, while

TABLE 1: The approximation ratio of the offline greedy user selection algorithm.

	Feeder, $f(\mu_{max}) = 24.5$				**Shanghai Traces,** $f(\mu_{max}) = 41.2$				**GeoLife,** $f(\mu_{max}) = 43.0$			
Budget	g-MUS	OPT	Ratio	Appr-Ratio	g-MUS	OPT	Ratio	Appr-Ratio	g-MUS	OPT	Ratio	Appr-Ratio
2	47.5	63.9	0.74	0.60	67.2	81.5	0.82	0.48	59.3	92.3	0.64	0.52
3	66.5	86.9	0.77	0.71	86.4	105.7	0.82	0.60	80.3	103.9	0.77	0.57
4	78.9	99.5	0.79	0.74	104.4	124.3	0.84	0.65	88.4	115.2	0.77	0.61
5	92.6	110.3	0.84	0.77	114.4	136.8	0.84	0.69	93.4	121.9	0.77	0.63



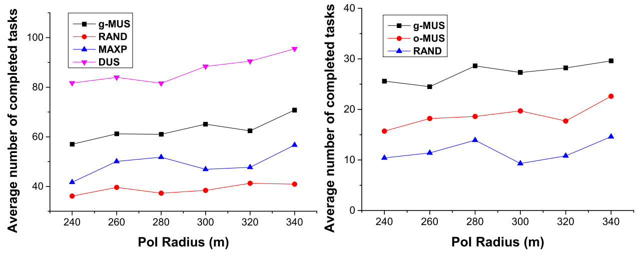
(a) $\mathcal{B} = 3, lifetime = 200$ (b) $lifetime = 200, m = 200$ (c) $\mathcal{B} = 3, m = 200$ (d) $\mathcal{B} = 3, lifetime = 200, m = 200$

(e) $\mathcal{B} = 3, lifetime = 200$ (f) $lifetime = 200, m = 200$ (g) $\mathcal{B} = 3, m = 200$ (h) $\mathcal{B} = 3, lifetime = 200, m = 200$

(i) $\mathcal{B} = 3, lifetime = 200$ (j) $lifetime = 200, m = 200$ (k) $\mathcal{B} = 3, m = 200$ (l) $\mathcal{B} = 3, lifetime = 200, m = 200$

Fig. 9: The simulation results of the *Feeder*, *Shanghai Trace*, and *GeoLife* in online case.

keeping the others fixed, as shown in Fig. 9. The simulation results show that o-MUS can achieve a high percentage of g-MUS and better than RAND in most of time, which shows the effectiveness of our proposed online user selection algorithm. Along with the increasing of variables, the completed tasks in o-MUS and g-MUS grow quickly and even have the similar trends, while the RAND rises but by a smaller increase. We also change the activetime of users in Fig. 9 (d, h, and l). When the activetime is short, there is an upward trend in the number of completed tasks. With the increase of activetime, our online algorithm cannot work so well. The reason is that the longer activetime is close to the duration of MCS campaign, which makes the users coming earlier have great advantages than the later ones, and thus, our online algorithm, which partitions the users into k segments by their coming orders and selects the best user in each segment, cannot work well. However, in reality, users won't work for a long time for MCS, so our assumption of the short activetime is usually reasonable.



(a) Offline scenario (b) Online scenario

Fig. 10: The relationship between the number of completed tasks and PoI radius.

Finally, we test the number of completed tasks along with the growth of PoI radius under the offline and online scenarios. As shown in Fig. 10, the number of completed tasks goes up slowly along with the growth of PoI radius. The reason is that the larger PoI radius may lead to a better prediction while the increase of the PoI radius is relatively small rather than the whole map. Meanwhile, we also test the average running times of the user selection algorithms in our prototype system on the Android platform. As shown in Table 2, both the online and offline algorithms achieve the short running times (less than 1s), which are totally acceptable in real-life deployments. In addition, the running

TABLE 2: The average running time of the user selection algorithms.

	Feeder		Shanghai Traces		GeoLife	
	MUS-D	MUS-C	MUS-D	MUS-C	MUS-D	MUS-C
offline (ms/user)	6.10	734.30	6.58	831.77	3.67	393.63
online (ms/user)	1.53	117.54	1.63	127.81	0.80	46.04

time of computing mobility prediction model for one day consumes around 5-10 minutes. Note that we only need to run it once for a period of time and also can obtain the computational help from the server if necessary.

9 Discussion

This section discusses issues that are not reported or addressed in this work due to space and time constraints, which can be added to our future work.

- *Spatiotemporal Data Correlation.* In this paper, we would like to recruit many users in order to cover all the tasks, which costs a lot and may even be impossible (*e.g.*, some tasks would have the remote locations or short lifetimes and no users can complete them). To deal with these problems, some researchers have proposed to exploit the spatiotemporal correlation between different tasks (*e.g.*, some nearby restaurants usually have the similar crowd flows) to complete a few tasks while intelligently inferring the others, which is called sparse MCS [27, 28]. In our future work, we plan to introduce the spatiotemporal data correlation into our framework, which can help reduce the required number of users and deal with the difficult tasks.
- *Privacy Protection.* In MCS, the privacy protection is very important and our proposed distributed user selection framework, called *Crowd UserS*, can protect privacy to a certain extent by using the distributed storage and computing architectures. However, it still needs some professional technologies or methods of privacy protection. For example, some researchers proposed to leverage the *Differential Geo-Obfuscation* to obfuscate the uploaded locations while achieving high sensing coverage [29, 30]. This method is suitable to be added into our user selection framework in the future work, where users can upload the obfuscated location to protect privacy and the server can use it to calculate the utilities according to the coverage.

10 Conclusion

In this paper, we discuss the user selection problem in H-MCS. We present the PoI based mobility prediction model to estimate the probabilities that spatial-temporal-sensitive tasks will be completed on time. Then, we propose a greedy offline user selection algorithm to select a approximately optimal user set under a participant number constraint. Furthermore, we extend our problem to a general online setting, and propose an online algorithm based on the offline one. Finally, we present the *Crowd UserS*, a distributed framework, and implement a prototype system. The extensive simulations have been conducted on three real-life mobile traces. The results prove the efficiency of our proposed *Crowd UserS* framework.

Acknowledgment

This work is supported by the the National Natural Science Foundation of China under Grant No. 61772230 and Natural Science Foundation of China for Young Scholars No. 61702215, China Postdoctoral Science Foundation No. 2017M611322 and No. 2018T110247, and Chinese Scholarship Council No. 201706170165. This work is also supported in part by NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, and CNS 1439672.

References

[1] W. Liu, Y. Yang, E. Wang, Z. Han, and X. Wang, "Prediction based user selection in time-sensitive mobile crowdsensing," in *IEEE SECON 2017 - IEEE International Conference on Sensing, Communication and Networking*, 2017.

[2] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[3] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, 2014.

[4] M. Karaliopoulos, O. Telelis, and I. Koutsopoulos, "User recruitment for mobile crowdsensing over opportunistic networks," in *IEEE INFOCOM 2015 - IEEE Conference on Computer Communications*, 2015.

[5] L. Pu, X. Chen, J. Xu, and X. Fu, "Crowd foraging: A qos-oriented self-organized mobile crowdsourcing framework over opportunistic networks," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[6] M. Xiao, J. Wu, H. Huang, L. Huang, and C. Hu, "Deadline-sensitive user recruitment for probabilistically collaborative mobile crowdsensing," in *IEEE International Conference on Distributed Computing Systems*, 2016.

[7] G. S. Tuncay, G. Benincasa, and A. Helmy, "Participant recruitment and data collection framework for opportunistic sensing: a comparative analysis," in *ACM MOBICOM Workshop on Challenged Networks*, 2013, pp. 25–30.

[8] Z. He, J. Cao, and X. Liu, "High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility." in *IEEE INFOCOM 2015 - IEEE Conference on Computer Communications*, 2015.

[9] A. Hassani, P. D. Haghighi, and P. P. Jayaraman, "Context-aware recruitment scheme for opportunistic mobile crowdsensing," in *The IEEE International Conference on Parallel and Distributed Systems*, 2015, pp. 266–273.

[10] D. Zhang, H. Xiong, L. Wang, and G. Chen, "Crowdrecruiter: Selecting participants for piggyback crowdsensing under probabilistic coverage constraint," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 703–714.

[11] Y. Liu, B. Guo, Y. Wang, W. Wu, Z. Yu, and D. Zhang, "Taskme: multi-task allocation in mobile crowd sensing," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016.

[12] H. Li, T. Li, and Y. Wang, "Dynamic participant recruitment of mobile crowd sensing for heterogeneous sensing tasks," in *IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2015, pp. 136–144.

[13] W. Gong, B. Zhang, and C. Li, "Task assignment in mobile crowdsensing: Present and future directions," *IEEE Network*, no. 99, pp. 1–8, 2018.

[14] Q. Yuan, I. Cardei, and J. Wu, "An efficient prediction-based routing in disruption-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 19–31, Jan 2012.

[15] E. Wang, Y. Yang, J. Wu, W. Liu, and X. Wang, "An efficient prediction-based user recruitment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 16–28, 2018.

[16] B. Guo, Y. Liu, W. Wu, Z. Yu, and Q. Han, "Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, pp. 392–403, 2017.

[17] J. Hamm, A. C. Champion, G. Chen, and M. Belkin, "Crowd-ml: A privacy-preserving learning framework for a crowd of smart devices," in *IEEE International Conference on Distributed Computing Systems*, 2015, pp. 11–20.

[18] Q. Xu and R. Zheng, "When data acquisition meets data analytics: A distributed active learning framework for optimal budgeted mobile crowdsensing," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017.

[19] D. Yang, G. Xue, G. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, pp. 1–13, 2015.

[20] E. Wang, Y. Yang, and L. Li, "A clustering routing method based on semi-markov process and path-finding strategy in dtn," *Chinese Journal Of Computers*, vol. 38, no. 3, pp. 483–499, 2015.

[21] L. Wang, D. Zhang, and H. Xiong, "Effsense: Energy-efficient and cost-effective data uploading in mobile crowdsensing," in *ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, 2013, pp. 1075–1086.

[22] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the Acm*, vol. 45, no. 4, pp. 634–652, 1999.

[23] M. Bateni, M. Hajiaghayi, and M. Zadimoghaddam, "Submodular secretary problem and extensions," *ACM Transactions on Algorithms (TALG)*, vol. 9, no. 4, p. 32, 2013.

[24] D. Zhang, J. Zhao, F. Zhang, R. Jiang, and T. He, "Feeder: Supporting last-mile transit with extreme-scale urban infrastructure data," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN '15. ACM, 2015, pp. 226–237.

[25] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[26] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on gps data," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 312–321.

[27] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[28] L. Wang, D. Zhang, D. Yang, A. Pathak, C. Chen, X. Han, H. Xiong, and Y. Wang, "Space-ta: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 2, p. 20, 2017.

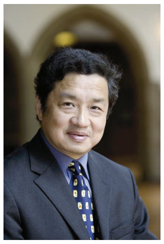
[29] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 627–636.

[30] W. Leye, Q. Gehua, Y. Dingqi, H. Xiao, and M. Xiaojuan, "Geographic differential privacy for mobile crowd coverage maximization." in *AAAI*, 2018.

Yongjian Yang received his B.E. degree in automatization from Jilin University of Technology, Changchun, Jilin, China in 1983; his M.E. degree in computer communication from Beijing University of Post and Telecommunications, Beijing, China in 1991; and his Ph.D. in software and theory of computer from Jilin University, Changchun, Jilin, China in 2005. He is currently a professor and a PhD supervisor at Jilin University, Director of Key lab under the Ministry of Information Industry, and Standing Director of the Communication Academy. His research interests include: network intelligence management, wireless mobile communication and services, and wireless mobile communication.

Wenbin Liu received his B.S. degree in Physics from Jilin University, Changchun, China in 2012; and M.E. degree in Department of Software from Jilin University, Changchun in 2016. He is currently a Ph.D. candidate in the Department of Computer Science and Technology, Jilin University, Changchun. His current research focuses on the Mobile CrowdSensing.

En Wang , the corresponding author, received his B.E. degree in software engineering from Jilin University, Changchun in 2011, and his M.E. degree and Ph.D. in computer science and technology from Jilin University, Changchun in 2013 and 2016. He is currently an associate professor in the Department of Computer Science and Technology at Jilin University, Changchun. His current research focuses on the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowdsensing.



Jie Wu is the Director of the Center for Networked Computing and Laura H.Carnell professor at Temple University. He also serves as the Director of International Affairs at College of Science and Technology. He served as Chair of Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Mobile Computing, IEEE Transactions on Service Computing, Journal of Parallel and Distributed Computing, and Journal of Computer Science and Technology. Dr. Wu was general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, ACM MobiHoc 2014, ICPP 2016, and IEEE CNS 2016, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.