When User Interest Meets Data Quality: A Novel User Filter Scheme for Mobile Crowd Sensing

Wensheng Li, Fan Li, Kashif Sharif
School of Computer Science, Beijing Institute of Technology
Beijing Engineering Research Center for High Volume Language Information Processing and Cloud Computing Applications
Beijing, China
{2120161011, fli, 7620160009}@bit.edu.cn

Yu Wang
Wireless Networking and Sensing (WiNS) Lab
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC, USA
yu.wang@uncc.edu

Abstract—Mobile crowd sensing has become a promising paradigm for mobile users to collect information. Considering that the task information push is not free and there are many users who are not interested in the current task or provide noisy sensing data, one of the imminent problems is how to recommend high-quality and interested users in real time and steer participators to collect data with adequate budgets. However, it is difficult to predict the data quality and users’ interest without the validity of real data. In this paper, we propose a user recommender system where the users’ data qualities for sensing tasks are derived from historical statistical data to filter out the non-interested and malicious users in current task. The aim is to recruit a sub-group of participators for efficient crowd sensing, in order to maximize the platform utility. We show that our problem is NP-hard, and model the recruitment process as a sub-modular problem. Finally, an approximation algorithm is designed to guarantee the platform utility and participators’ profits. We evaluate our algorithm on simulated data set and the results indicate that the platform utility and data quality improves significantly.

I. INTRODUCTION

Recent years have witnessed rapid proliferation of mobile devices. These devices are equipped with various embedded sensors, such as accelerometer, gyroscope, camera, global position system (GPS) and finger print reader to acquire environmental information, and have enabled powerful computation and communication capabilities for diverse applications. The large quantity and advanced functionalities of smart mobile devices have created a new paradigm, Mobile Crowd Sensing (MCS) [1]–[3], which involves individuals using smart devices to monitor human activities and sense environmental information related to their interested phenomenons, which is then sent to the cloud platform through 4G networks and WiFi for advanced treatment.

Although some promising approaches have been developed, mobile crowd sensing still has many challenges. Mobile users decide whether to participate in the sensing task according to their interests. When starting to perform a sensing task, mobile users consume their resources such as energy and memory. Besides, mobile users might have potential privacy concerns by sharing their personal data. For these reasons, mobile users would not be interested or devoted in participating in mobile crowd sensing tasks voluntarily. Thus, it is necessary to design an incentive mechanism to motivate mobile users to participate in crowd sensing tasks [4]–[7]. Furthermore, due to the existence of malicious users in mobile crowd who execute the tasks poorly, the data uploaded to the platform may be noisy or completely unusable [8]–[10]. This reduces the quality of data and creates an irreversible loss for platform profits. Besides, the cloud platform issues many MCS tasks, and users are not interested in each task. Considering the network overhead and the consumption of user device resource, the platform needs to bear the cost of mobile network operators and the related users for the information push as shown in Fig. 1. It is unproductive to send the task information push to the non-interested and malicious users, because the non-interested users will not give responses and the involvement of malicious users will affect the quality of the final sensing data. For example, in the application of urban monitoring [11], [12], we may sense the Points Of Interest (POIs) within the region, so we launch a recruitment among interested and reputable users only. Hence, a large amount of blind information push will increase the platform cost, and receive few valid responses.

To solve this problem, a user recommender system is required, for selecting suitable users from raw mobile users to send the sensing task information push, and then allocate and put forward an incentive mechanism for the recruitment to ensure the reward of platform. In reality, even without considering the final user selection, designing an efficient and effective user recommender system in crowd sensing applications is a tricky problem, and most previous studies have processed this kind of problems based on ranking of
users. The platform sorts users according to the response times of tasks in [13], [14] and then decides which users to deliver information push. However, it is not fair for users who have never participated in any crowd sensing applications and are suitable for the current application. But because there is no historical data, they will not be selected to receive information push. Moreover, this approach is not conducive to eliminate the malicious users who can raise their rankings by different response strategies and ultimately limit the choices of platform.

In addition, many applications recruit users with posted pricing incentives [15] which may lead to the user’s heterogeneous income, because participants will feel uncomfortable if they participate in the sensing task with high cost. Therefore, a more reasonable solution is to allow users to make their own budgets, and the platform will make a strategy to select the appropriate participants after gathering all the users’ budgets [12], [15], [16]. It is interesting to note that the scheme of users’ unsolicited offer is more universally used, and can ensure the interests of users. Consequently, a large number of crowd sensing and crowd sourcing systems currently adopt autonomous pricing to encourage users participation.

Surprisingly, although autonomous pricing is a very common incentive scheme to encourage users, the fundamental problem of how to steer users to make reasonable budgets is still intractable. If the autonomous price is much higher than the cost, the total revenue of the platform will decrease, and it will force the platform to select users with poor data qualities.

In this paper, we study a user recommender system to ensure the data quality and the task interestingness in MCS to reduce the cost of information push. In this system, sensing tasks and participants are described by some properties, and the information of data qualities and users’ cost can be obtained form the historical sensing data. Our aim is to pick up the data quality and the task interestingness in MCS to reduce the cost, the total revenue of the platform will decrease, and it will force the platform to select users with poor data qualities.

The remainder of this paper is organized as follows. We first discuss related work in Section II. We describe the crowd sensing model and present our problem formulation in Section III. We then present our information push mechanism and allocation algorithm along with an incentive mechanism in Section IV and V, respectively. Section VI shows the proof of the user budget truthfulness. Section VII evaluates the performance and Section VIII concludes this paper finally.
that each user is relatively rational, that is, the submitted budget is higher than the actual cost, which also applies to malicious users.

In this paper, we consider that the sensing data from different users is the same for the platform, except the data quality. If the quality of the sensing data is very high or low, changes in unit quality have little impact on platform value. Conversely, if the data quality is relatively moderate, changes have a significant impact. This assumption is very common in daily activities, for example, the work where a student improves the grade from 99 to 100 is harder than the work where student increase the grade from 49 to 50. So we set the mapping function \( g(i) \) from the data quality to the original value by the user \( i \) to the platform as

\[
g(i) = \zeta \cdot \frac{1}{1 + e^{-\frac{4}{5} \cdot \frac{q(i) - \sigma}{\zeta}\times5}},
\]

where \( q(i) \) is the data quality of user \( i \), \( q(i) \in (0, 2\sigma) \), \( \sigma \) means the median of data quality, and \( \zeta \) is the highest value a user creates for the platform. \( 1/(1 + e^{-z}) \) satisfies the value change curve we need when \( z \in (-5, 5) \).

Considering that each user only know their own budget, and that there is a kind of potential competition relationship between mobile users, the users do not share their relevant information with each other, and therefore, we can not guarantee that each sensing candidate can submit an optimal budget. The value of sensing data is different for platform if the user is selected as a participator at different time. As the number of selected users increases, the value of the same user will decrease to the platform, i.e., for any selected groups \( X \subseteq Y \) and any sensing candidate \( x \notin Y \), we have \( v(x|X) \geq v(x|Y) \), where \( v(x|X) = v(X \cup \{x\}) - v(X) \) and \( v(X) \leq v(Y) \). This function is called sub-modular. According to formula (2), the real value function created by user \( i \) is

\[
v(i) = v(i) \times S(X) \times g(i),
\]

where \( S(X) \in (0, 1) \) represents the attenuation function of the selected group to the current value, and with the increase in the number of selected participators \( X \), the value of \( v(i) \) drops faster. In the following sections, we will use these characteristics to solve our problem.

### IV. USER FILTER BASED ON RECOMMENDER SYSTEM

In our solution, the first step is to pick the appropriate users from the raw user group on the basis of the current task, which involves the interactive information and data quality information between sensing tasks and users. We set up a recommender system for learning and prediction. In order to do this, we perform statistical analysis on the data which records historical task-user participation.

We define \( A \) as the tasks set, and \( U \) to represent the users set. In the crowd sensing application, the platform has published \( K \) types of tasks, so \( K = |A| \), where each type of task is described by the feature vector \( A_k \), and vector \( A_k \) has \( M \) dimensions. The \( m-th \) dimension describes the value of the task on the feature space \( R_m = \{r_{m1}, r_{m2}, \ldots, r_{ml}\} \).
and $A_{km} \in R_m$, so the feature vector of task $k$ is $\bar{A}_k = (A_{k1}, A_{k2}, \ldots, A_{kM})$.

The platform has $N = |U|$ registered users, and each user is described by a feature vector called $\bar{U}_i$. Similar to the feature vector $A_k$, $\bar{U}_i$ also has $C$ dimensions, and each dimension is evaluated on the feature $S_c = \{s_{c1}, s_{c2}, \ldots, s_{cn}\}$, so $U_{ic} \in S_C$ and the feature vector of user $i$ is $\bar{U}_i = (U_{i1}, U_{i2}, \ldots, U_{ic})$.

We define the task-user relationship matrix $F_{K \times N}$ and $F(t,i)$ to represent the relationship between user $i$ and task $t$. $F(t,i)$ has four types of values as shown in formula (4). If $F(t,i) = -3$, it indicates that the platform has never sent the information push of task $t$ to user $i$. If $F(t,i) = -2$, it represent user $i$ didn’t submit its budget of task $t$ to the platform in the history. $F(t,i) = -1$ means although user $i$ has submitted a budget in history for sensing task $t$, platform did not select him as a final participator. Besides, $F(t,i) = \theta$, where $\theta \in (0, 2\sigma)$ symbolizes the quality of user $i$ in task $t$. The specific form is as follow,

$$F(t,i) = \begin{cases} -3, & \text{if } i \notin X_t; \\ -2, & \text{if } i \in X_t \text{ and no response; } \\ -1, & \text{if } i \in X_t \text{ but } i \notin P_t; \\ \theta, & \text{if } i \in P_t. \end{cases} \tag{4}$$

When task $t$ is ready to be released, our aim is to send the information push to the users who ensures the interest and the data quality in the current task, so it is necessary to adopt a recommendation strategy to filter out the non-interested and malicious users from the platform. First of all, according to the task-user relationship matrix $F$ and the predefined malicious data quality threshold $\gamma$, we classify the user $i$ into three categories (non-interested, malicious, preferred) according to the value $F(t,i)$, and record each ratio of the three categories according to the different values in each dimension of the user’s feature vectors. After that, we get the percentage of the total non-interested and malicious users and users. Then we record each value of the user’s description features. When we focus on the feature $a$ in the user feature space, if the ratio of in-valid user is greater than or equal to the threshold $\mu$, we will filter out all users who has the feature $a$ from raw platform registered users. Thus, we can get a preliminary user base by performing above operation on each of user features according. We call this method a content-based recommender system.

After the content-based recommender system, the cost of information push will reduce, but it does not guarantee that the rest of users will provide quality data for task $t$, as there may still be some known malicious users and potential malicious users. Known malicious users who have poor data quality record in the historical data, but they were not filtered out from the group by using content-based recommender system. For these known malicious users in task $t$, we can filter them out directly. But for potential malicious users, considering that they have not previously participated in the sensing task $t$ and that they don’t have any properties marked in the first step, we cannot find them just with the relationship of task $t$.

The specific form is as follow,

$$f(t,i) = \begin{cases} -3, & \text{if } i \notin X_t; \\ -2, & \text{if } i \in X_t \text{ and no response; } \\ -1, & \text{if } i \in X_t \text{ but } i \notin P_t; \\ \theta, & \text{if } i \in P_t. \end{cases} \tag{4}$$

When task $t$ is ready to be released, our aim is to send the information push to the users who ensures the interest and the data quality in the current task, so it is necessary to adopt a recommendation strategy to filter out the non-interested and malicious users from the platform. First of all, according to the task-user relationship matrix $F$ and the predefined malicious data quality threshold $\gamma$, we classify the user $i$ into three categories (non-interested, malicious, preferred) according to the value $F(t,i)$, and record each ratio of the three categories according to the different values in each dimension of the user’s feature vectors. After that, we get the percentage of the total non-interested and malicious users and users. Then we record each value of the user’s description features. When we focus on the feature $a$ in the user feature space, if the ratio of in-valid user is greater than or equal to the threshold $\mu$, we will filter out all users who has the feature $a$ from raw platform registered users. Thus, we can get a preliminary user base by performing above operation on each of user features according. We call this method a content-based recommender system.

Algorithm 1: User Recommendation WorkFlow

```
Input: F, A, U, M, C, \gamma, \mu, \lambda, \text{and task } t
Output: X_t

1 // Initialization:
2 ft(U) = \emptyset, X_t \leftarrow U;
3 for i = 1, 2, \ldots N do
4   for c = 1, 2, \ldots C do
5     if U_{ic} \notin ft(U) then
6       ft(U) = ft(U) \cup \{U_{ic}\};
7       S_{pf}(U_{ic}) \leftarrow 0;
8       S_{iv}(U_{ic}) \leftarrow 0;
9     end
10   end
11 for i = 1, 2, \ldots N do
12   for c = 1, 2, \ldots C do
13     if F(t,i) \geq \gamma then
14       S_{pf}(U_{ic}) \leftarrow S_{pf}(U_{ic}) + 1;
15     end
16     else if 0 \leq F(t,i) < \gamma or F(t,i) = -2 then
17       S_{iv}(U_{ic}) \leftarrow S_{iv}(U_{ic}) + 1;
18     end
19   end
20 for i in ft(U) do
21   if S_{iv}(U_{ic}) \geq \mu then
22     remove all users have property $i$ from $X_t$;
23   end
24 for i = 1, 2, \ldots K do
25   if \rho(A_i, A_t) \geq \lambda then
26     remove all malicious users in task $i$ from $X_t$;
27   end
28 return $X_t$
```

In the second step, we adopt a neighborhood-based filtering method. According to the description features of tasks, we set the similarity function of task feature vector as

$$\rho(t, x) = \frac{\sum_{i=1}^{M} L(A_{ti}, A_{xi})}{M}, \tag{5}$$

where $M$ is the number of tasks’ feature dimension, $t \in A$, $x \in A$, and $L(a, b)$ is an indicating function, which can be defined as

$$L(a, b) \begin{cases} 1, & \text{if } a=b; \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We use the formula (5) and formula (6) to calculate the feature similarity between each task in task set and the task $t$. We record the malicious users from the tasks where the value of $\rho(t, \cdot) \geq \lambda$. Because of the high similarity between tasks, and many users who have not participated in the task $t$ have
historical data qualities in other tasks, we probably believe these users have similar performances in task $t$. Therefore, we can filter out these users who have low data qualities in similar tasks from the preliminary users. The detailed user recommendation work flow is shown in Algorithm 1.

Algorithm 1 first records all the feature attributes of the registered users and initializes the number of preferred and invalid users by the feature attributes (line 2-11), then record the number of preferred and invalid users by the feature attributes according the historical data quality (line 12-20). Finally, it removes the invalid users by recommender system (line 22-26, 27-31).

V. INCENTIVE ALLOCATION AND FINAL SELECTION MECHANISM

The user recommender system provides the best possible candidates for information push of task $t$. We now design an incentive mechanism to encourage candidates to participate in sensing task $t$, and to steer candidates to submit reasonable budgets to the best possible extent. The budgets follow the Gauss distribution in theory, and most of candidates make rational offers based on their own costs. However, considering the privacy of candidates, their own profits, and the potential competitive relationship between candidates, candidates don’t share their bidding information with others. There is no absolute strategy to ensure that each candidate makes the right budget. Special effort is required to steer users to submit relatively rational budgets. In this paper, we define the user's payment function as

$$ u_t(i) = b_t(i) + \alpha S(x)[v_t(i) - b_t(i)], $$

where the payment $u_t(i)$ consists of two part: the user budget $b_t(i)$ and the bonus of profits created by user $i$, and $\alpha$ is a parameter factor. In the Section VI, we theoretically analyze this incentive method to ensure the user budget is truthful according to the formula (7).

At the final user selection part, we design an approximate approach to recruit users. After receiving all candidates' budgets, platform calculates the utility created by each candidate based on the data quality and the user's budget. For the candidates without historical data qualities in sensing task $t$, platform will give them a default data quality $\gamma$. And then, platform will arrange candidates in descending order according to the expected utility, and select final users until the platform's utility stops increasing. Algorithm 2 describes the selection process in detail. We calculate the utilities created by users for the platform (line 3-11), and we uses a greedy strategy to select the user (line 13-16).

We adjust the user's payment function slightly based on the real sensing data and the value. If the real data quality is greater than the expected data quality, user will get more prizes, and if user's real value is less than the budget, we will pay the user in proportion to the value. Finally, we update the task-user relationship matrix $F$. If user $i$ is a new participant, we use real data quality directly, otherwise, we adjust the value of $F(t, i)$ according to a certain weight. Formula (8) and (9) describe the user $i$ payment function and data quality update function in sensing task $t$, respectively.

$$ u_t(i)' = \begin{cases} u_t(i) + \beta(v_t(i)' - v_t(i)), & \text{if } v_t(i) < v_t(i)'; \\ b_t(i) \cdot v_t(i)' / v_t(i), & \text{if } v_t(i)' < b_t(i); \\ u_t(i) + \alpha S(X)(v_t(i)' - v_t(i)), & \text{otherwise}. \end{cases} $$

Where $u_t(i)'$, $v_t(i)'$ represent real user $i$ payment and data value, respectively, and $\beta$ is an extra bonus parameter.

$$ F(t, i) = \begin{cases} F(t, i)', & \text{if } F(t, i) \notin (0, 2\sigma); \\ (1 - \omega)F(t, i) + \omega F(t, i)', & \text{otherwise}. \end{cases} $$

Where $F(t, i)'$ represents the real data quality by user $i$ in sensing task $t$, and $\omega$ is a weight of $F(t, i)'$.

VI. ANALYSIS OF UFBC

In this section, we prove that the user budget guarantee truthfulness in UFBC, and user can not get more payment by raising his budget maliciously.

Lemma 1. Budget $b_t(i)$ submitted by user $i$ is truthful.

Proof. For convenience, we assume that the expected data quality of user $i$ is the same as the real data quality, and the original value of user $i$ is $g(i)$, so the payment of user $i$ is $u(i) = b(i) + \alpha S(X)[S(X)g(i) - b(i)]$. After collecting all users budget information, the platform can determine the final recruitment set $C$ and the selection rank $\{l_1, l_2, ..., l_{|C|}\}$. For any user $i$ in the final selected group, the payment is $u(i) \geq b(i)$. The last selected user's budget $b(l_{|C|})$ is
hence, we get $S(C)$ as

$$S(C) = \frac{b(l_{[C]}l)}{g(l_{[C]}l)},$$

(11)

user $l_i$ is selected when the selected group is $C_i \subset C$, so the payment $u(l_i)$ is

$$u(l_i) = b(l_i) + \alpha S(C_i)[S(C_i)g(l_i) - b(l_i)].$$

(12)

If the user $l_i \in C$ tries to increase the budget from $b(l_i)$ to $\hat{b}(l_i)$, we perform the following case analysis.

**Case 1:** The user $l_i$ is not selected if the budget $\hat{b}(l_i) > S(C_i)g(l_i)$, and it will not be recruited, so the payment $v(l_i) = 0$.

**Case 2:** We can get the budget $\hat{b}(l_i)$ upper bound, which should be less than or equal to $S(C_i)g(l_i)$ if and only if the user $l_i$ is selected at the last position, so the payment $\hat{u}(l_i)$ can be described as

$$\hat{u}(l_i) = \hat{b}(l_i) + \alpha S(C)[S(C)g(l_i) - \hat{b}(l_i)],$$

(13)

and $\hat{u}(l_i)$ should be greater than $u(l_i)$ according to the formula (12) and (13). We can get the budget $b(l_i)$ lower bounds, which should be greater than

$$\hat{b}(l_i) > \frac{b(l_i)[1 - \alpha S(C_i)] - \alpha g(l_i)[S^2(C_i) - S^2(C)]}{1 - \alpha S(C)} = b(l_i).$$

(14)

In order to ensure $\hat{b}(l_i) > b(l_i)$, we should set the $\hat{b}(l_i)$ as

$$\hat{b}(l_i) > \max(b(l_i), \hat{b}(l_i)),$$

(15)

so, we can get the value range of $\hat{b}(l_i)$, and user $l_i$ will get a higher payment when the budget is within the value range. It is not realistic for any user to change the budget within the range as the user does not know the budget information of other users, and does not know the accurate rank in selection list. Besides, user risks the possibility of not to participate in the sensing task if it raises the budget.

Overall, user will not raise the budget blindly to get more payment, and in order to ensure that it can be selected by the platform, the user budget will be as close as possible to the actual cost. So this incentive mechanism can make sure the truthfulness of the user budget.

**VII. Performance Evaluation**

This section evaluates the UFBC with numerical simulation from different aspects, including the utility of the platform, quality of data received, effect of historical task similarity in dataset, and the user’s payment.

In simulated data, we divide registered users into new users, non-interest users, malicious users, regular users and the users who have never been selected in the current task. The real data quality, the expected data quality and the budget of each user are simulated by different forms of Gauss distribution according to the user’s category. UFBC system parameters, are set as shown in **TABLE II**. The user’s feature vector has 6 dimensions, and each dimension has 6 different values. Similarly, each task is described by a 5 dimensional vector with 3 different values per dimension. About the simulated data, we set the ratio of new, non-interested, non-responsive, malicious and regular users are 5%, 25%, 2%, 8%, 60% respectively. The data quality of malicious user $i$ is $q(i) \sim N(3.5, 1)$, the data quality of preferred user $j$ is $q(i) \sim N(6, 1)$, and the budget of interested user $l$ is $b(i) \sim N(60, 100)$.

Based on formula (1), we implemented two baseline algorithms donated by Pure Greedy (PG), and Content-Based Recommender (CBR). PG selects users according to the budgets from low to high; and CBR selects users according to the utility created by each user from the group that filters out the known non-interested and malicious users in the current task. Each of the following experiments has been repeated 200 times.

We set the number of the platform register users from 0 to 3000 with the increment of 100, and we compare UFBC with the above-mentioned two benchmark algorithms for the utility of the platform, as shown in Fig. 2. With the increase in the number of users, the utility of the platform increases gradually, but the growth rate decreases. We find that the utility of UFBC is better than the other two algorithms in the long run. The result shows that, when the size of registered users is relatively small, the impact of data quality is minimal for the platform utility because the number of recruitments is less than demand, and the user interest is the dominant factor in the platform interest. But with the increase in the number of registered users, the existence of malicious users and the cost of information push leads to a reduction in the platform utility in PG, and CBR. UFBC is able to stabilly maintain good performance. Due to the influx of large numbers of users, the ability of CBR to filter in-valid users is reduced, and some known and potential malicious users infiltrate the final selected group, which leads to an increase in push cost, and resultanty in decline of data quality and platform utility.

We also analyze the real data quality of all the selected users. As we expected, the average data quality in UFBC is always better than the other two benchmark methods in Fig. 3. With the increase in number of users, the average data quality in UFBC is always greater than the data quality threshold $\gamma = 5.0$. When the number of users is large enough, the

**TABLE II**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha)</td>
<td>0.3</td>
</tr>
<tr>
<td>(\beta)</td>
<td>0.5</td>
</tr>
<tr>
<td>(\mu)</td>
<td>0.7</td>
</tr>
<tr>
<td>(\lambda)</td>
<td>0.8</td>
</tr>
<tr>
<td>(\delta)</td>
<td>1.0</td>
</tr>
<tr>
<td>(\gamma)</td>
<td>5.0</td>
</tr>
<tr>
<td>(\sigma)</td>
<td>5.0</td>
</tr>
<tr>
<td>(\zeta)</td>
<td>200.0</td>
</tr>
<tr>
<td>(S(X))</td>
<td>((1 - \frac{|X|}{2000})^2)</td>
</tr>
</tbody>
</table>
average of data quality in UFBC returns to the mean value of regular users. Data quality in PG algorithm is erratic, as the user selection is solely based on the budget bid. CBR performs better in contrast, but inability to filter all types of users which may degrade the data quality hampers its overall performance, which is lower than UFBC.

A major objective of UFBC is to reduce the push cost of system by selection of appropriate users. Fig. 4 shows the cost of information push against the user base size. Due to refined selection in UFBC, the cost to disseminate task information is very less compared to other algorithms.

To analyze the influence of different values of task similarity $\lambda$ for the final platform utility, we set $\lambda$ from 1.0 to 0.0 with the decrement of 0.2, and we set the number of registered users $|U| = 3000$. From Fig. 5 we can see that the platform utility rises first and the decreases to 0 with the decrease of task similarity $\lambda$. The platform can get a higher utility by setting a suitable value of $\lambda$. Fig. 6 also reflects the same behavior. With the decrease of $\lambda$, the expected and real data quality continues to increase, and we observe that there will be at most one user hired for the task when $\lambda \leq 0.4$. This also explains why data quality is smaller than the threshold. The effect of $\lambda$ only creates an insignificant difference between expected and real data quality in UFBC. Because the simulated data is not the same as the real data, we believe that this phenomenon is more evident in the real world. As the historical data set will grow and task similarity and data quality of users increases, the accuracy of the algorithm will also increase.

We also test the user’s payment under different budgets. We verify the individual payment by randomly picking a valid user (ID=1137) and increase the budget of user 1137 from 0 to the maximum task value $\zeta_1 = 200$ in the task $t=1$ when the other parameters and data qualities are the same. We illustrate the result in Fig. 7. The budget-payment relationship, with the increase of budget, the payment of user 1137 will increase first and then decrease. When the budget is $\geq 49$, user 1137 will not be selected as a final participant, so user can not raise his budget blindly to increase his payment. Considering that a high budget leads to a reduction of the user's priority of selection, the user will choose a lower budget to obtain the priority of selection under the same payment. The sudden drop in rank-line in Fig. 7 at budget value of 49 does not signify and improvement in rank, but rather the dropout of user from
the selection.

VIII. CONCLUSION

Selection of participants in a crowdsourcing task is as important as utility optimization. Effective recruitment strategy in real time generates not only good quality data, but also increases the system and participant utility. In this paper, we propose a novel user filter scheme for mobile crowd sensing, based on the historical task info to maximize the platform utility. Given a sensing task, UFBC can send the information push to the specific groups to ensure the interest and the data quality of the group. It uses content-based recommender system to filter out the users who have unsuited user feature in current task from the user base, and neighbor-based recommender system to further filter out the remaining known and potential malicious user. We also design an incentive mechanism that motivates mobile users to submit rational budgets. The analysis and simulation results indicate the effectiveness of our approach, as the utility of platform is higher for larger user base and the budgets of users remain realistic, while returning good quality data.

ACKNOWLEDGMENTS

The work of Fan Li is partially supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61772077, 61370192 and 61432015. The work of Yu Wang is partially supported by the US National Science Foundation under Grant No. CNS-1319915 and CNS-1343355, and NSFC under Grant No. 61572347. F. Li is the corresponding author.

REFERENCES