2020 17th IEEE International Conference on Sensing, Communication and Networking (SECON)

Combinatorial Multi-Armed Bandit Based User Recruitment in Mobile Crowdsensing

Hengzhi Wang
Computer Science and Technology
Jilin University
Changchun, China
wanghz17@mails.jlu.edu.cn

Yongjian Yang
Computer Science and Technology
Jilin University
Changchun, China
yyj@jlu.edu.cn

En Wang*
Computer Science and Technology
Jilin University
Changchun, China
wangen@jlu.edu.cn

Wenbin Liu
Computer Science and Technology
Jilin University
Changchun, China
liuwb16@mails.jlu.edu.cn

Yuanbo Xu
Computer Science and Technology
Jilin University
Changchun, China
yuanbox@jlu.edu.cn

Jie Wu
Computer and Information Sciences
Temple University
Philadelphia, USA
jiewu@temple.edu

***Abstract*—Mobile Crowdsensing (MCS) is a new paradigm that recruits users to cooperatively perform a sensing task. When recruiting users, existing works mainly focus on selecting a group of users with the best objective ability, e.g., the user's probability or frequency of covering the task locations. However, we argue that, for the cooperative MCS task, the completion effect depends not only on the user's objective ability, but also on their subjective collaboration likelihood with each other. In other words, in each single round, we prefer to recruit users with not only a strong objective ability but also good collaboration likelihood. Moreover, even though we could find a well-behaved group of users in a single round, in the multi-round scenario without enough prior knowledge, we still face the problem of recruiting previously well-behaved user groups (exploitation) or recruiting unknown user groups (exploration). To address these problems, in this paper, we first convert the single-round user recruitment problem into the min-cut problem and propose a graph theory based algorithm to find the optimal group of users. Furthermore, in the multi-round scenario, to balance the trade-off between exploration and exploitation, we propose the multi-round User Recruitment strategy based on the combinatorial Multi-armed Bandit model (URMB) and prove that it can achieve a tight regret bound. Finally, extensive experiments on three real-world datasets validate that the users recruited by URMB result in a better task completion effect than the state-of-the-art strategy.**

***Index Terms*—Mobile crowdsensing, user recruitment, collaboration likelihood, combinatorial multi-armed bandit, min-cut**

I. Introduction

With the proliferation of smart devices equipped with powerful sensors, Mobile CrowdSensing (MCS) has attracted much attention [1], which recruits distributed mobile users to cooperatively perform various sensing tasks [2], [3] such as air quality monitoring [4], traffic mapping construction [5] and target tracking [6]. Thus, how to select users raises the fundamental user recruitment problem in MCS.

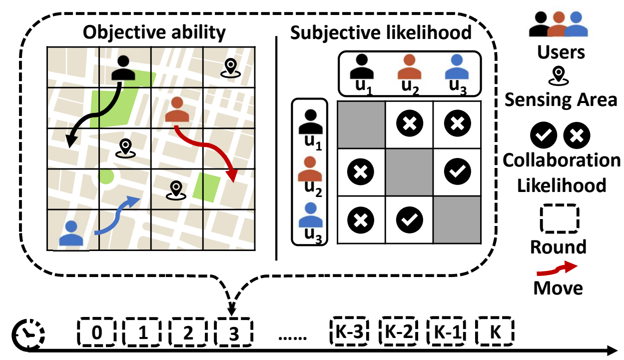


Fig. 1: An example to illustrate the multi-round user recruitment problem taking both objective ability and subjective collaboration likelihood into consideration.

The existing user recruitment strategies mainly focus on selecting a group of users with the best objective ability, e.g., the user's probability or frequency of covering the task locations [2], [7]. However, in this paper, we argue that, for the cooperative MCS tasks, the completion effect depends not only on the objective ability, but also on their subjective collaboration likelihood with each other. For example, the target tracking task requires a group of users to detect and upload the target's location information as well as the detecting time. However, the uploading process also leaks the detector's spatiotemporal information. If a user is not familiar with the others in this group, it may be unwilling to cooperate with them because of privacy leaks [8]. Thus, even if the user finds the target, it may hesitate about whether to upload the location, which results in the poor task completion effect. Hence, when recruiting users for these cooperative tasks, we

This work is supported by the National Natural Science Foundations of China under Grant No. 61772230 and No. 61972450, Natural Science Foundation of China for Young Scholars No. 61702215, China Postdoctoral Science Foundation No. 2017M611322 and No. 2018T110247, Changchun Science and Technology Development Project No.18DY005, and in part by NSF grants CNS 1824440, CNS 1828363, CNS 1757533, CNS 1629746, CNS 1651947, and CNS 1564128.

*En Wang is the corresponding author.

978-1-7281-6630-8/20/$31.00 2020 © IEEE

should consider not only their objective abilities but also the subjective collaboration likelihood with each user.

To clearly describe the user recruitment problem in this paper, an example is shown in Fig. 1, where the whole recruitment process is divided into K rounds. There are three users (u_1, u_2, and u_3) moving around the locations of the sensing task. At the beginning of each round, some users are selected to perform the sensing task, and the users' objective abilities (as shown in the left part of Fig. 1) are defined as the frequencies that they pass through the locations of the sensing task. Meanwhile, the subjective collaboration likelihood is shown in the right part of Fig. 1, which shows that u_2 and u_3 are willing to cooperate with each other, while they do not want to cooperate with u_1. In this single-round scenario, we face the problem of selecting a group of users who have strong individual objective ability as well as good collaboration likelihood, i.e., the single-round user recruitment problem. Furthermore, in the multi-round scenario, because we are unknown to users' actual objective ability and subjective collaboration likelihood, we must estimate them according to the feedbacks (the completion effect of the sensing task) of previous rounds. Even though we could estimate the ability and collaboration likelihood of users, we still face the problem of continuing recruiting previously well-behaved user groups or exploring other unknown user groups. This is exactly the multi-round user recruitment problem to be addressed.

Taking the subjective collaboration likelihood into consideration, we cannot individually measure users' abilities anymore. In other words, a user's actual ability depends not only on its objective ability, but also on its cooperators within the same group. Thus, how to *model each user's actual ability to finish a sensing task* is the first challenge. Moreover, since a user's collaboration likelihood is unknown, when a user is added to a group, the amount of utility change for this group is hard to measure (maybe non-linearly or non-monotonically). Hence, the previous greedy-liked recruiting strategies [2], [7] do not work anymore. So, in the single-round scenario, how to *recruit a group of users with both a strong objective ability and good collaboration likelihood* is the second challenge. Finally, in the multi-round scenario without enough prior knowledge, a conservative strategy is to continue recruiting the previously well-behaved users. By contrast, a progressive strategy is to try recruiting unknown users. Thus, how to *balance the trade-off between exploration and exploitation in the multi-round scenario* is the third challenge.

In order to overcome the above challenges, we first formulate the user ability in a graph, where the vertex indicates the user's objective ability and the edge indicates the collaboration likelihood with each other. Second, we convert the single-round user recruitment problem into the min-cut problem [9], and select the best group of users based on the modified min-cut algorithm for undirected graph. Third, we propose the multi-round user recruitment strategy based on the combinatorial multi-armed bandit model (URMB) [10], [11] to balance exploration and exploitation. Moreover, we prove that the URMB can achieve a tight regret bound through theoretical analysis. Our main contributions are summarized as follows:

- As we know, this is the first work considering not only users' objective ability, but also their subjective collaboration likelihood with each other when recruiting users in MCS.
- In each single round, we convert the user recruitment problem into the min-cut problem, and propose a graph theory based algorithm to find the optimal solution. We also prove the equivalence of the conversion, and the completeness and optimality of the proposed algorithm.
- For the multi-round user recruitment problem without enough prior knowledge, we propose the multi-round user recruitment strategy based on combinatorial multi-armed bandit model (URMB) to balance exploration and exploitation. We also prove that the proposed strategy achieves a tight regret bound.
- We conduct extensive simulations based on three real-world datasets to verify the performance of the proposed strategy, and the results show that URMB always outperforms other strategies.

II. Related Work

A. User Recruitment in MCS

User recruitment problem in MCS has been studied for years [2], [3], [12]. When recruiting users, some researches often consider the user's objective ability, such as the ability of covering sensing tasks [2], [7]. Gao $et\ al.$ [2] consider the joint probability of multiple vehicles performing the tasks and propose the winner selection algorithm based on the non-trivial set cover problem. Yang $et\ al.$ [7] investigate the prediction-based user recruitment framework based on the optimal stopping theory. However, both of them assume that the user's ability is known in advance. Wu $et\ al.$ [13] consider the situation where the users' abilities are unknown, and develop a Thompson Sampling based user selection algorithm. Gao $et\ al.$ [14] propose an extended UCB based unknown user recruitment process. But, [13] and [14] ignore each user's subjective collaboration likelihood with others.

B. Combinatorial Multi-armed Bandit Problem

To balance exploration and exploitation, the multi-armed bandit problem (MAB) has been studied for years, and there are several well-known algorithms: ϵ-greedy algorithm [15], upper confidence bound algorithm [16] and so on. Gai $et\ al.$ [10] further extend the MAB to the combinatorial multi-armed bandit problem (CMAB) where multiple variables (arms) can be selected at each time. They investigate the linearly weighted combination of selected variables and propose the LLR algorithm with the bounded regret. For solving the CMAB problem, Chen $et\ al.$ [11] propose a general algorithm CUCB with a large of nonlinear reward instances. Hüyük $et\ al.$ [17] investigate the combinatorial Thompson Sampling strategy and achieve a better regret than [11]. However, these algorithms for CMAB could not be directly used to solve the user recruitment problem in MCS since when they calculate the reward of a group of arms, they ignore the arm's influence on each other.

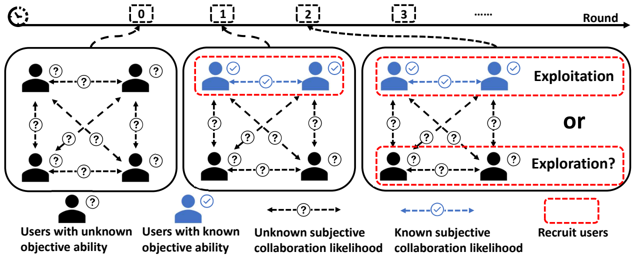


Fig. 2: The illustration for exploration and exploitation.

III. System Model and Problem Formulation

A. System Model

We consider a general MCS system, where a cooperative sensing task τ is published by the platform in some sensing areas at the beginning and will remain there until the end of system time. Users are denoted as $U = \{u_1, \cdots, u_M\}$, each of them is constantly moving around sensing areas. We adopt the frequency of u_i passing through the sensing areas in round t to denote its objective ability ρ_i^t, and all objective abilities of users in U are represented as the vector $\boldsymbol{\rho^t}$. Note that the objective ability of each user follows a fixed distribution during a long-term time, which will be verified in our experiments. Moreover, we adopt the likelihood matrix $\boldsymbol{\alpha} = (\alpha_{11}, \alpha_{12}, \cdots, \alpha_{ij}, \cdots, \alpha_{MM})$ to denote the collaboration likelihood between any two users in U, and $\alpha_{ij} \in [0, 1]$ denotes the collaboration likelihood between u_i and u_j. For example, if u_i and u_j are familiar, the collaboration likelihood between them is high, which leads to a good task completion effect since the familiar users have a strong willingness to cooperate. If they are unfamiliar, the collaboration likelihood between them is low, then the completion effect is bad. Obviously, $\alpha_{ij} = \alpha_{ji}$ due to the symmetry of the collaboration likelihood, and α_{ii} will not be used.

The MCS campaign is divided into K rounds represented as $T = \{t_1, \cdots, t_K\}$. In a single round t, the platform selects a group of users S^t to perform the cooperative task. For a cooperative task, the completion effect depends not only on the user's objective ability, but also on their collaboration likelihood with each other. To measure the task completion effect of S^t, we consider the quality of data (QoD) [18] of task τ, i.e., the number of times the task is performed, which is represented as follows:

$$Q^t(S^t) = \sum_{u_i \in S^t} \bar{\alpha}_i^t \cdot \rho_i^t, \tag{1}$$

where $\bar{\alpha}_i^t = \sum_{u_j \in S^t, j \neq i} \alpha_{ij}^t / (|S^t| - 1)$ denotes the average collaboration likelihood among u_i and other users in S^t. ρ_i^t and α_{ij}^t are the objective ability of u_i and the collaboration likelihood between u_i and u_j. Then, the platform selects users per round until the end of K rounds. For the multi-round scenario, the total QoD of K rounds is $\sum_{t \in T} Q^t(S^t)$.

B. Problem Formulation

In the single round t, the single-round user recruitment problem is formulated as follows:

$$\text{Maximize} \quad Q^t(S^t) \tag{2}$$

$$\text{Subject to} \quad |S^t| = N. \tag{3}$$

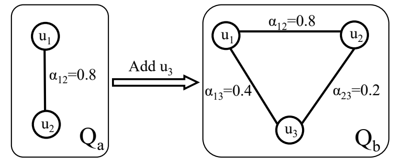


Fig. 3: An example for single-round recruitment problem.

The constraint (3) indicates that the platform selects N users in a round because of some resource constraints, e.g., the budget constraint [2]. Selecting a fixed number of users in each round is a special case of our problem, we further use simulations in Section VI to prove that selecting a variable number of users by our algorithm could still achieve satisfactory performance.

Furthermore, in the multi-round scenario without enough user prior knowledge, how to maximize the total QoD after all rounds is the multi-round user recruitment problem, which is formulated as follows:

$$\text{Maximize} \quad \sum_{t \in T} Q^t(S^t) \tag{4}$$

$$\text{Subject to} \quad |S^t| = N, \ \forall t \in T. \tag{5}$$

To maximize the total QoD after K rounds, a conservative strategy is to continue recruiting the previously well-behaved users, while a progressive strategy is to try recruiting unknown users. An example of the trade-off between exploration and exploitation is shown in Fig. 2, where the blue users indicate the known users with accurate objective ability and the black users are the unknown users with inaccurate objective ability. The dashed lines indicate the collaboration likelihood between any two users. When selecting users, we should decide to explore more unknown users or exploit more known users.

IV. User Recruitment with Objective Ability and Collaboration Likelihood

We focus on the user recruitment problem in single-round and then multi-round scenarios.

A. Single-Round Strategy

We first consider the single-round scenario, where we select a group of users S based on their objective abilities and collaboration likelihoods for the current round t. As shown in Eq. 2, the single-round user recruitment problem is non-linear and non-monotonic, since the $\bar{\alpha}_i, \forall u_i \in S$ will change when a new user is added into the group S. An example is shown in Fig. 3, assume that the objective ability of each user $\rho_1 = \rho_2 = \rho_3 = 2$ and $\alpha_{12} = 0.8$, $\alpha_{13} = 0.4$, $\alpha_{23} = 0.2$. Then, we can obtain the QoD $Q_a = 3.2$ according to Eq. 2. However, after adding u_3 to S, we have the $Q_b = 3 < 3.2$. Therefore, it is difficult to find the best group of users to maximize the utility function, and existing greedy-liked user recruitment strategies [2], [7] cannot work effectively anymore. Moreover, if we search for the optimal solution globally, we need to search $\binom{M}{N}$ times in the worst case, which leads to a very high time complexity $O(M^N)$. Obviously, searching globally for the best group is not a good choice. Thus, to find the best group of users efficiently, we convert the single-round user recruitment problem into the min-cut problem in graph theory through the following steps:

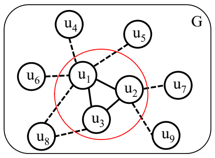


Fig. 4: An example for constructed graph G.

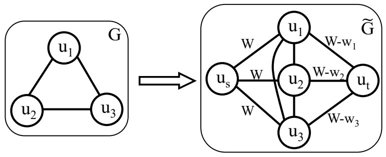


Fig. 5: An example for the conversion from G to $\tilde{G}$.

First step: Without loss of generality, we construct an edge weighted undirected graph $G = (V, E)$, where each vertex represents a user, denoted as $V = U = \{u_1, \cdots, u_i, \cdots, u_M\}$. Moreover, for the edge between two vertices u_i and u_j, the weight is calculated based on the abilities of them:

$$w_{ij} = \frac{(\rho_i + \rho_j) \cdot \alpha_{ij}}{N-1}, \forall u_i, u_j \in U, i \neq j. \tag{6}$$

Note that G is a complete graph because we should consider the collaboration likelihood between any two users. An example of a constructed graph G is shown in Fig. 4, and we only demonstrate a part of the edges for brevity.

Then, the single-round user recruitment problem in Eq. 2 turns into the problem of finding the induced subgraph $G' = (V', E')$ with the maximum edge weight as follows:

$$\text{Maximize} \quad \sum_{u_i, u_j \in V', i \neq j} w_{ij} \tag{7}$$

$$\text{Subject to} \quad |V'| = N. \tag{8}$$

Second step: As shown in Fig. 4, assume that the subgraph covered by the red circle is the optimal subgraph $G' = (V', E')$ with the maximum edge weight, then the total weight of the solid edges is equal to $Q(S), S = V'$ in Eq. 2 according to Eq. 6. In other words, as soon as we find these solid edges, the single-round user recruitment problem is solved. However, it is difficult to find these edges directly, so we tend to find the edges which are connected to the vertices in V' and are not included in E', i.e., the dashed edges in Fig. 4. Furthermore, we find that the dashed edges are cut by the red circle, which is the cut $c(V', V \setminus V')$ that divides V into two disjoint parts V' and $V \setminus V'$. We call these edges cut by $c(V', V \setminus V')$ cut-set, and the weights are $w(c(V', V \setminus V')) = \sum_{u_i \in V'} \sum_{u_j \in V \setminus V'} w_{ij}$.

Definition 1 (u_s-u_t Cut)**.** A u_s-u_t cut is a partition of vertex set V into two vertex sets, S and T, and $u_s \in S$, $u_t \in T$.

Definition 2 (Minimum Cut)**.** The minimum cut is the cut with minimum $w(c(S, T))$.

Let $w_i = \sum_{u_j \in V, j \neq i} w_{ij}$ represent the weight of all edges between u_i and all vertices in V. We have

$$\sum_{u_i, u_j \in V', i \neq j} w_{ij} = \frac{1}{2} \cdot \Big(\sum_{u_i \in V'} w_i - w(c(V', V \setminus V'))\Big). \tag{9}$$

Then, we turn the maximization problem in Eq. 7 into the minimization problem in Eq. 11 as follows:

$$\text{Maximize} \quad \frac{1}{2} \cdot \Big(\sum_{u_i \in V'} w_i - w(c(V', V \setminus V'))\Big) \tag{10}$$

$$\Leftrightarrow \text{Minimize} \quad w(c(V', V \setminus V')) - \sum_{u_i \in V'} w_i \tag{11}$$

$$\text{Subject to} \quad |V'| = N. \tag{12}$$

Algorithm 1: Minimum Cut

Input: Graph $G = (V, E)$
Output: cut $c(S', V \setminus S')$

1 $w_0 \leftarrow +\infty$;
2 **for** each $u_i \in \{u \in V | u \neq u_s, u \neq u_t\}$ **do**
3 $\quad S \leftarrow \{u_i\}$;
4 $\quad$ **while** $|S| < N + 1$ **do**
5 $\quad\quad$ Search the vertex a such that $w(S, a) = \max\{w(S, b) | b \in V \setminus S, b \neq u_t\}$;
6 $\quad\quad S \leftarrow S \cup \{a\}$;
7 $\quad$ **if** $w(c(S, V \setminus S)) \leq w_0$ **then**
8 $\quad\quad w_0 \leftarrow w(c(S, V \setminus S))$;
9 $\quad\quad S' \leftarrow S$;
10 **return** cut $c(S', V \setminus S')$

For the $\sum_{u_i \in V'} w_i$ in Eq. 11, we regard $-w_i$ as the cost of adding vertex u_i into the subgraph G'. Afterwards, we build a sink vertex u_t and connect all vertices to it, and assign $-w_i$ to the edge between u_i and u_t as $w_{it} = -w_i$. Due to the non-negative property of edge weight, we add a large enough variable $W = \sum_{u_i \in V} w_i$ to the weight $w_{it} = W - w_i$. In this way, based on the graph theory [9], [19], we can convert the graph $G = (V, E)$ to a new undirected graph $\tilde{G} = (\tilde{V}, \tilde{E})$ through three steps: (1) add source vertex u_s and sink vertex u_t to the vertex set V; (2) connect all vertices in V to source u_s (undirected edge) and assign the weight W; (3) connect all vertices in V to sink u_t (undirected edge) and assign the weight $W - w_i$. This process can be formulated as follows:

$$\tilde{V} = V \cup \{u_s, u_t\}, \tag{13}$$

$$\tilde{E} = \{(i,j) | (i,j) \in E\} \cup \{(u_s, u_i) | u_i \in V\} \cup \{(u_i, u_t) | u_i \in V\}, \tag{14}$$

$$w_{si} = W, \quad u_i \in V, \tag{15}$$

$$w_{it} = W - w_i, \quad u_i \in V. \tag{16}$$

An example of the conversion process is shown in Fig. 5. Through the conversion, the problem of finding the maximum weight subgraph in G turns to be the problem of finding the minimum u_s-u_t cut in $\tilde{G}$. The equivalence of these two problems will be proved in Section V.

Next, we propose a specific minimum cut algorithm as shown in Algorithm 1 based on the undirected graph min-cut algorithm [20]. In Algorithm 1, we first initialize the temporary variable w_0 as a positive infinity number (line 1). Then, for each vertex u_i in $V \setminus \{u_s, u_t\}$, we set u_i as the initial vertex for set S (line 3). Next, the algorithm enters the loop phase until $|S| = N + 1$, where N is the number of the selected vertices as shown in constraint (12). In the loop phase (lines 4~6), we select the vertex which is connected to the vertex set S most tightly from the vertex set $V \setminus S$ (line 5) and add it to S (line 6), where $w(S, u_j) = \sum_{u_i \in S} w_{ij}$ is defined as the weights of all edges between the vertex set S and vertex u_j. Moreover, we find the minimum cut $c(S', V \setminus S')$ (lines 7~9), which will be proved to be the minimum u_s-u_t cut. Therefore, as long as we input the converted graph $\tilde{G}$ to Algorithm 1, we can get the minimum u_s-u_t cut of $\tilde{G}$. Note that the solution to the problem in Eq. 7 is the vertex set $V' = S' \setminus \{u_s\}$, and

Algorithm 2: Update Among Rounds

Input: All the observed data $(\boldsymbol{\rho_o^t}, Q_o^t), t \in [0, m]$
Output: Updated likelihood matrix $\boldsymbol{\alpha^{m+1}}$

1 $\rho_i^{t+1} \leftarrow \sum_{r=1}^{k(t)} \rho_{i,r}/k(t)$;
2 **while** $\eta \frac{\partial J(\boldsymbol{\alpha^m})}{\partial \alpha_{ij}^m} \leq \varepsilon, \forall \alpha_{ij}^m$ in $\boldsymbol{\alpha^m}$ **do**
3   **for** each α_{ij}^m in $\boldsymbol{\alpha^m}$ **do**
4         $\alpha_{ij}^m \leftarrow \alpha_{ij}^m - \eta \frac{\partial J(\boldsymbol{\alpha^m})}{\partial \alpha_{ij}^m}$;
5 $\boldsymbol{\alpha^{m+1}} \leftarrow \boldsymbol{\alpha^m}$;
6 **return** $\boldsymbol{\rho^{m+1}}, \boldsymbol{\alpha^{m+1}}$

the solution to single-round user recruitment problem in Eq. 2 is exactly the users in V'.

In conclusion, we solve the single-round user recruitment problem in Eq. 2 through the following steps: (1) construct a graph $G = (V, E)$ based on users' objective abilities and their collaboration likelihoods, and convert the single-round user recruitment problem to the maximum weight subgraph problem 8; (2) construct a new graph $\tilde{G}$ through the process in Eqs. 13-16, and convert the subgraph problem to the minimum u_s-u_t cut problem in the undirected graph; (3) propose an algorithm based on graph theory to solve the min-cut problem.

B. Update Strategy

After solving the single-round user recruitment problem, we focus on the multi-round user recruitment problem. Note that the single-round scenario uses the deterministic user information to select users. However, in the multi-round scenario without enough prior knowledge, we don't know users' objective abilities and collaboration likelihoods and have to learn them during the multiple rounds. Thus, before describing the multi-round user recruitment problem, we introduce the update strategy. In fact, at the end of each round, the objective ability (the frequency of each selected user passing through the task) and the actual QoD (the number of times the task is performed) are observable, i.e., feedbacks. Thus, we can update users' abilities based on the feedbacks of previous rounds.

Update on objective ability: Assume that user u_i has been selected $k(t)$ times before round t, and the actual objective ability observed at the r-th is $\rho_{i,r}$. Then, the user's objective ability in round $t+1$ is updated as follows:

$$\rho_i^{t+1} = \frac{\sum_{r=1}^{k(t)} \rho_{i,r}}{k(t)}, \tag{17}$$

Update on collaboration likelihood: We update the estimated likelihood matrix based on the batch gradient descent (BGD) [21]. Assume that the current round is the m-th round. First, we define the hypothetical function $\hbar$ of any round t and the loss function J at the end of round m according to Eq. 1:

$$\hbar(\boldsymbol{\alpha^t}, \boldsymbol{\rho^t}) = Q^t(S^t) = \sum_{u_i \in S^t} \bar{\alpha}_i^t \cdot \rho_i^t, \tag{18}$$

$$J(\boldsymbol{\alpha^m}) = \frac{1}{2m} \sum_{t=0}^{m} (\hbar(\boldsymbol{\alpha^m}, \boldsymbol{\rho_o^t}) - Q_o^t)^2, \tag{19}$$

where $\boldsymbol{\rho^t}$ and $\boldsymbol{\alpha^t}$ denote the estimated objective ability vector and likelihood matrix in round t respectively. Note that the loss function $J(\boldsymbol{\alpha^m})$ denotes the total loss of all previous rounds at the end of round m, and $\hbar(\boldsymbol{\alpha^m}, \boldsymbol{\rho_o^t})$ denotes the expected value of QoD based on the actual observed objective ability vector $\boldsymbol{\rho_o^t}$ at the end of round t and the estimated likelihood matrix $\boldsymbol{\alpha^m}$ of current round m. Moreover, Q_o^t denotes the observed value of the QoD at round t. Then we propose the update strategy as shown in Algorithm 2. First, we need to input the observed data of all rounds before round m (including), i.e., $\boldsymbol{\rho_o^t}$ and $Q_o^t, t \in [0, m]$. Line 1 represents the update on the objective ability vector according to Eq. 17. Then we enter the loop and update the value of each item α_{ij}^m in the likelihood matrix $\boldsymbol{\alpha^m}$ (lines 3-4), where $\frac{\partial J(\boldsymbol{\alpha^m})}{\partial \alpha_{ij}^m}$ is the gradient of loss function J, and η denotes the step size (learning rate). The loop stops when the descent distances of all items are less than the accuracy ε (line 2). Finally, we can get the updated likelihood matrix $\boldsymbol{\alpha^{m+1}}$, which will be used for the user recruitment process in the next round.

Algorithm 3: Multi-Round User Recruitment

Input: User set U, user objective ability vector $\boldsymbol{\rho}$, likelihood matrix $\boldsymbol{\alpha}$
Output: Total QoD Q after K rounds

$Q \leftarrow 0$, $t \leftarrow 0$, $\boldsymbol{\rho^t} \leftarrow \boldsymbol{\rho}$, $\boldsymbol{\alpha^t} \leftarrow \boldsymbol{\alpha}$;
$r_i \leftarrow 1$ for each $u_i \in U$;
while $t \leq K$ **do**
    $t \leftarrow t + 1$;
    for each ρ_i^t *in* $\boldsymbol{\rho^t}$ **do**
        $\tilde{\rho}_i^t \leftarrow \rho_i^t + \sqrt{\frac{3 \ln t}{2 r_i}}$;
    Convert the user set U to graph G through Eq. 6 with $\boldsymbol{\tilde{\rho}^t}$, $\boldsymbol{\alpha^t}$;
    Convert graph G to graph $\tilde{G}$ through Eqs. 13-16;
    $S' \leftarrow$ Minimum Cut $(\tilde{G})$;
    Select the users in $S' \setminus \{u_s\}$;
    for each $u_i \in S' \setminus \{u_s\}$ **do**
        $r_i \leftarrow r_i + 1$;
    Users perform the task and we observe the actual values of $\boldsymbol{\rho_o^t}$ and Q_o^t at the end of round t;
    $\boldsymbol{\rho^{t+1}}$, $\boldsymbol{\alpha^{t+1}} \leftarrow$ Update Among Rounds ();
    $Q \leftarrow Q + Q_o^t$;
return Q

C. Multi-Round Strategy

Now, we combine the above single-round algorithm and update strategy and propose the multi-round user recruitment strategy based on the combinatorial multi-armed bandit model (URMB). In the multi-round scenario, the platform doesn't know each user's accurate objective ability and collaboration likelihood. In order to maximize the total QoD after K rounds, the platform should try to learn each user's ability according to the feedbacks of the previous rounds as accurately as possible (exploration), and recruit those users who are currently estimated to have a strong ability (exploitation). Neither pure exploration nor pure exploitation will bring the best results.

Therefore, how to balance exploration and exploitation is the key of the multi-round user recruitment problem.

We propose Algorithm 3 based on CUCB [11] to solve the problem. First, we input the user set U and a little prior knowledge about each user's ability $\boldsymbol{\rho}$ and $\boldsymbol{\alpha}$ for the first recruitment, which actually determines the regret bound of the algorithm. After the initialization process (lines 1-2), the algorithm enters the loop when $t \leq K$ (line 3). In each round t, we face the single-round user recruitment problem. Note that the objective ability vector $\boldsymbol{\rho^t}$ used in the single-round user recruitment strategy (lines 7-9) includes an adjustment term $\sqrt{3 \ln t / 2r_i}$ (line 6) to balance exploration and exploitation, where t indicates the number of rounds and r_i denotes the number of times u_i is selected in the previous rounds. Moreover, we regard the prior knowledge about users' objective abilities as if each user is selected once, and thus we initialize $r_i = 1$ for each user (line 2). Furthermore, we select users in $S' \setminus \{u_s\}$ to perform task τ and update r_i (lines 11-12). Then, at the end of each round, both the actual objective ability of each user (the frequency of passing through the task) and the number of the task is performed are observable. Next, we update the selected users' abilities based on the update strategy and get the latest user ability for the next round (line 14). Note that the update strategy needs all observed data up to the current round t. Finally, we add the observed QoD Q_o^t of the current round to the total QoD Q.

V. Theoretical Analysis

For the conversion process from graph G to $\tilde{G}$ in Eqs. 13-16, we convert the problem of finding the maximum weight induced subgraph G' of graph G to the problem of finding the minimum u_s-u_t cut $c(S, V \setminus S)$.

Lemma 1. Any u_s-u_t cut $c(S,T)$ of the converted graph $\tilde{G}$ one-to-one corresponds to a subgraph G' of graph G.

Proof. Assume that $c(S,T)$ is a u_s-u_t cut of $\tilde{G}$, $u_s \in S$, $u_t \in T$. Since the cut is a partition of $\tilde{G}$ into S and T, we can always find only one subgraph $G' = (V', E')$ of G such that $V' = S \setminus \{u_s\}$. $\square$

Theorem 1. The cut-set of cut $c(S', V \setminus S')$ found by Algorithm 1 is the solution to the problem in Eq. 7.

Proof. The problem in Eq. 7 is to find a subgraph $G' = (V', E')$ to maximize $\sum_{u_i,u_j \in V', i \neq j} w_{ij}$. Assume that the cut $c(S,T)$ in $\tilde{G}$ is found by Algorithm 1, according to Lemma 1, we have $V' = S \setminus \{u_s\}$ and $V \setminus V' = T \setminus \{u_t\}$. Thus, we get

$$\begin{aligned} w(c(S,T)) &= \sum_{u_i \in S, u_j \in T} w_{ij} \\ &= \sum_{u_i \in V'} w_{it} + \sum_{u_j \in V \setminus V'} w_{sj} + \sum_{u_i \in V', u_j \in V \setminus V'} w_{ij} \\ &= |V| \cdot W + \sum_{u_i \in V'} (-w_i + \sum_{u_j \in V \setminus V'} w_{ij}) \\ &= M \cdot W - \sum_{u_i, u_j \in V', i \neq j} w_{ij}. \end{aligned} \tag{20}$$

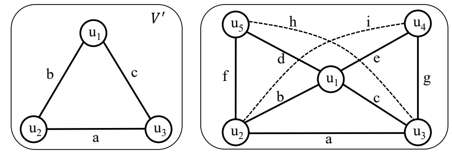


Fig. 6: Illustration for optimality analysis.

The last equation holds because w_i represents the weights of all edges between u_i and all other vertices in V, and w_{ij} denotes the weights of all edges between u_i and all other vertices in $V \setminus V'$, and G' is the induced subgraph. Thus, $-w_i + \sum_{u_j \in V \setminus V'} w_{ij} = -\sum_{u_j \in V'} w_{ij}$. Moreover, $M \cdot W$ is a constant term in Eq. 20. Thus, when finding the minimum u_s-u_t cut, we get the vertex set $V' = S \setminus \{u_s\}$ as well as the maximum $\sum_{u_i,u_j \in V', i \neq j} w_{ij}$, which is the solution to the problem in Eq. 7. $\square$

Theorem 2. Algorithm 1 can find at least a u_s-u_t cut $c(S,T)$ of the converted graph $\tilde{G} = (\tilde{V}, \tilde{E})$ and $|S \setminus \{u_s\}| = N$.

Proof. In the conversion process from G to $\tilde{G}$, we assign the $W = \sum_{u_i \in V} w_i$ to the edge between u_s and any vertex in V. Thus, $w_{is} = W > w_{ij}, \forall u_j \in \tilde{V} \setminus u_s$. Then, in Algorithm 1 (lines 5), the first searched vertex must be u_s, so u_s will be added into S. Meanwhile, the limitation in the search process $b \neq u_t$ ensures that $u_t \notin S$. Thus, the cut is a u_s-u_t cut. $\square$

Theorem 3. The solution founded by Algorithm 1 is the optimal solution to the problem in Eq. 7.

Proof. According to Theorems 1 and 2, we know Algorithm 1 can find a solution to the problem in Eq. 7, then we will prove the optimality of the solution $V' = S \setminus \{u_s\}$. Moreover, since the weights of edges between u_s and any vertex u_i except u_t are equal, we ignore vertex u_s in the following process.

When $|V'| = 2$, suppose that the two vertices in the optimal set V' are u_a and u_b. Furthermore, assume that when the initial vertex is u_a (line 3 in Algorithm 1), the next added vertex is u_c. We have $w_{ac} \geq w_{ab}$ according to line 5, which contradicts the hypothesis that u_a and u_b are the optimal set. Thus, the next added vertex is u_b and Algorithm 1 finds the optimal set.

When $|V'| = 3$, assume that the optimal set is $V' = \{u_1, u_2, u_3\}$, and the corresponding edge weights are a, b and c as shown in Fig. 6. Assume that $a \geq b \geq c$. For the worst case, if Algorithm 1 does not find V' in the process when u_2 and u_3 are initial vertices (line 3), then we will prove that V' will be found when u_1 is the initial vertex. Assume that in the first iteration in lines 4∼6, the initial vertex is u_2 and the next added vertex is an arbitrary vertex u_5, then we have $f > a$, let $f = a + \varepsilon_1$. Since the optimal set V' are u_1, u_2 and u_3, then the weight of $\{u_1, u_2, u_5\}$ is $f + b + d < a + b + c$ and we get $d < c - \varepsilon_1$. For the same reason, the weight of $\{u_2, u_3, u_5\}$ is $a + f + h < a + b + c$, then $h < b + c - a - \varepsilon_1$. Similarly, in the second iteration, if the initial vertex is u_3 and the next added vertex is an arbitrary vertex u_4, let $g = a + \varepsilon_2$, we get $e < b - \varepsilon_2$ and $i < b + c - a - \varepsilon_2$. Subsequently, when the initial vertex is u_1, the first vertex added to S must be u_2 because $w(S, u_2) = b >$

(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 7: The distribution of check-ins of three datasets.

$w(S,u_3)=c>w(S,u_5)=d$ and $b>w(S,u_4)=e$. Now $S=\{u_1,u_2\}$ and the next added vertex must be u_3 because that $w(S,u_3)-w(S,u_4)=a+c-(e+i)>2(a-b+\varepsilon_2)>0$ and $w(S,u_3)-w(S,u_5)=a+c-(f+d)>0$. Therefore, Algorithm 1 will find the optimal user set V'.

Similarly, when $|V'|>3$, in Algorithm 1, once the initial vertex is determined $S\leftarrow\{u_i\}$, the second added vertex, e.g., u_j is determined due to line 5. Specifically, we can combine u_i and u_j into a single vertex u_k such that $w_{kx}=w_{ix}+w_{jx},\forall u_x\in V$ in the following adding vertex process, i.e., $S=\{u_i,u_j\}\rightarrow S^*=\{u_k\}$. It does not affect the subsequent algorithm process because $w(S,u_x)=\sum_{u_y\in S}w_{yx}=w_{ix}+w_{jx}=w_{kx}=w(S^*,u_x)$. Consequently, the case of $|V'|>3$ can be converted into the case of $|V'|=3$, whose optimality has been proven. In conclusion, the optimality is proved. $\square$

Lemma 2. For an arbitrary user set S, the expected value of QoD is monotonically non-decreasing with the objective ability vector $\boldsymbol{\rho}$, i.e., $\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})\le\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}')$ if $\rho_i\le\rho_i',\forall u_i\in S$.

Proof. The monotonicity is obvious according to Eq. 18. $\square$

Lemma 3. For an arbitrary user set S, $\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})$ in Eq. 18 achieves bounded smoothness such that for any two objective ability vectors $\boldsymbol{\rho}$ and $\boldsymbol{\rho}'$, if $\max_{u_i\in S}|\rho_i-\rho_i'|\le\vartheta$, where ϑ is a constant, we have $|\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})-\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}')|\le\hbar(\boldsymbol{\alpha},\boldsymbol{\vartheta})$, where $\boldsymbol{\vartheta}$ indicates a new objective ability vector such that $\rho_i=\vartheta,\forall u_i\in S$.

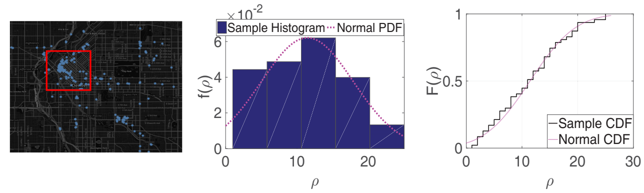
Proof. Since $\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})=\sum_{u_i\in S}\bar{\alpha}_i\cdot\rho_i$, if $\max_{u_i\in S}|\rho_i-\rho_i'|\le\vartheta$, we have $|\hbar(\boldsymbol{\alpha},\boldsymbol{\rho})-\hbar(\boldsymbol{\alpha},\boldsymbol{\rho}')|=\sum_{u_i\in S}\bar{\alpha}_i\cdot|\rho_i-\rho_i'|\le\sum_{u_i\in S}\bar{\alpha}_i\cdot\vartheta=\hbar(\boldsymbol{\alpha},\boldsymbol{\vartheta})$, proved. $\square$

Theorem 4. The multi-round recruitment strategy achieves the regret of Algorithm 3 in K rounds at most $(\frac{6\ln K}{f^{-1}(\Delta_{min})^2}+\frac{\pi^2}{3}+1)\cdot\Delta_{max}\cdot\Delta_{ratio}$.

Proof. Suppose that the function $f(\boldsymbol{\rho})=\hbar(\boldsymbol{\alpha^0},\boldsymbol{\rho})$, where $\boldsymbol{\alpha^0}$ denotes the initial priori knowledge of likelihood matrix between users. Thus, $f(\boldsymbol{\rho})$ achieves monotonicity and bounded smoothness due to lemmas 2-3. According to [11], we have

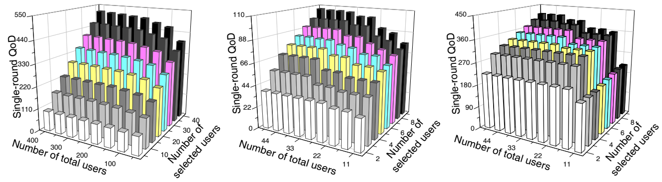
$$\begin{aligned}Reg(K)=&K\cdot opt-\sum_{t=0}^{K}Q^t(S^t)\le K\cdot opt-\Delta_{ratio}\cdot\sum_{t=0}^{K}f(\boldsymbol{\rho^t})\\ \le&(\frac{6\ln K}{f^{-1}(\Delta_{min})^2}+\frac{\pi^2}{3}+1)\cdot\Delta_{max}\cdot\Delta_{ratio}.\end{aligned}\tag{21}$$

$\Delta_{min}=opt-\max_{S^t}\{Q^t(S^t)\}$, $\Delta_{max}=opt-\min_{S^t}\{Q^t(S^t)\}$ and $\Delta_{ratio}=\min_{S^t}\{\frac{\hbar(\boldsymbol{\alpha^t},\boldsymbol{\rho^t})}{f(\boldsymbol{\rho^t})}\}$ for $t\in[0,K]$, where opt denotes the QoD of the optimal user set S_{opt} with the real likelihood matrix and S^t is the user set selected in round t. When $K\rightarrow+\infty$, $Reg(K)$ approaches a constant. $\square$



(a) Distrubution (b) PDF (c) CDF

Fig. 8: Test for normal distribution.



(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 9: QoD evaluation in the single-round scenario.

VI. Performance Evaluation

A. Datasets and Settings

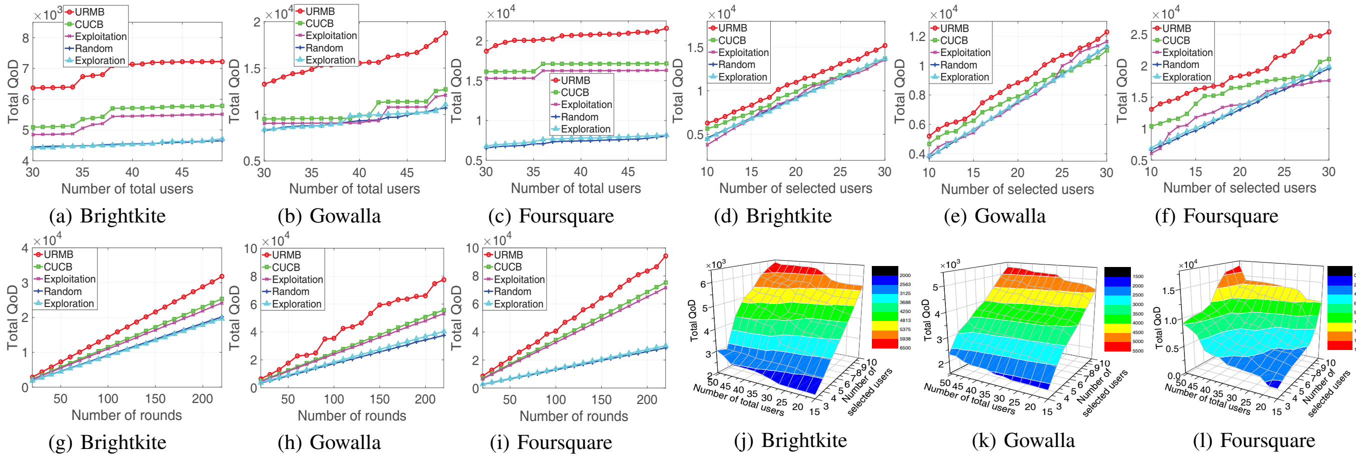
In the simulations, we adopt three widely-used real-world datasets, i.e., Brightkite [22], Gowalla [22] and Foursquare [23]. Theses three datasets contain the user's location and social information, thus they are suitable for simulations. Brightkite includes 58228 nodes (users), 214078 edges (collaboration likelihood) and a total of 4491143 check-ins over the period of Apr. 2008 - Oct. 2010. Similarly, Gowalla consists of 196,591 nodes, 950,327 edges and 6,442,890 check-ins. Foursquare contains 456988 check-ins made by 10162 users. As shown in Fig. 7, the blue nodes denote the geographic distribution of three datasets and red circles represent the areas with concentrated check-ins.

We introduce the settings of simulations as follows: (1) The number of rounds K: we divide check-ins of all users into K rounds by date. (2) User objective ability vector $\boldsymbol{\rho^t}$: the frequency of each user passing through the sensing area during round t represents its objective ability in round t, which is actually the observed value of the user objective ability at each round. (3) Real likelihood matrix $\boldsymbol{\alpha_r}$: the real likelihood matrix is adopted to generate the observation data Q_o^t in Eq. 19 according to Eq. 1. For two users u_s and u_t, if they are friends in the dataset, then we quantify the likelihood α_{ij} into a random number between 0.5 and 1. If not, we quantify the likelihood α_{ij} into a random number between 0 and 0.5. (4) Initial likelihood matrix $\boldsymbol{\alpha_0}$: initial matrix represents the prior knowledge, which is generated randomly between 0 and 1. (5) The number of total users M: for each dataset, we only keep the users with at least 1000 check-ins. (6) The number of the selected users N: it can change from 1 to M.

B. Baselines and Metrics

We mainly compare URMB with following algorithms:

- *Optimal*: It searches globally to find the optimal solution with each user's real ability.
- *CUCB* [11]: It both considers exploration and exploitation when selecting users with the initial likelihood matrix $\boldsymbol{\alpha_0}$, and takes Algorithm 1 as its Oracle part.



(a) Brightkite (b) Gowalla (c) Foursquare (d) Brightkite (e) Gowalla (f) Foursquare

(g) Brightkite (h) Gowalla (i) Foursquare (j) Brightkite (k) Gowalla (l) Foursquare

Fig. 10: QoD evaluation in the multi-round scenario.

TABLE I: Running time of three datasets

Running time (ms)	N (URMB)				N (Optimal)			
	2	3	4	5	2	3	4	5
Brightkite	8	7	8	9	4	7	43	1278
Gowalla	10	10	9	9	4	6	50	685
Foursquare	30	10	9	8	3	5	52	642

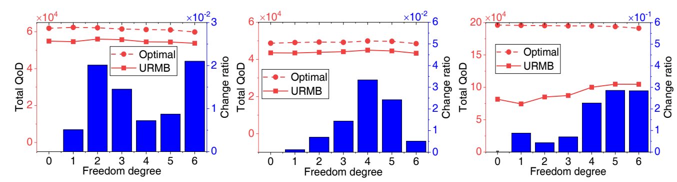
- *Exploitation*: It only considers the current ability of each user when selecting users, i.e., Algorithm 3 without the adjustment term in line 6.
- *Exploration*: It only selects N users with the least number of selections.
- *Random*: It selects N users randomly.

We use following metrics to evaluate the compared algorithms: (1) QoD Q_s for the single-round scenario. (2) Total QoD Q for the multi-round scenario. (3) Running time for algorithms. (4) Freedom degree d_f, which represents the degree of the actual number of selected users N' deviates from the specified number of selected users N, i.e., $N' \in [N - d_f, N + d_f]$. (5) The loss J in Eq. 19.

C. Evaluation Results

Before evaluating the performance of the URMB, we first verify that users' objective abilities (frequencies of passing through the sensing areas) in different rounds follow a normal distribution as shown in Fig. 8. We adopt the 2094 check-ins of a user in Brightkite as shown in Fig. 8a (Denver), where the red rectangle indicates the sensing area, and get the user's objective ability vector $\boldsymbol{\rho}$ for all rounds. The frequency histogram graph and the PDF of the normal distribution with the mean and standard deviation of $\boldsymbol{\rho}$ are shown in Fig. 8b, and the CDFs are shown in Fig. 8c. Clearly, the objective ability of a user follows a stable distribution so that URMB can learn the mean of each user's objective ability through Eq. 17.

QoD for single-round scenario: Based on three datasets, we conduct simulations of the single-round scenario ($K = 1$), and the results are shown in Fig. 9. For Brightkite in Fig. 9a, when M changes from 50 to 450 and N changes from 5 to 40, QoD for single scenario Q_s shows an upward trend, which



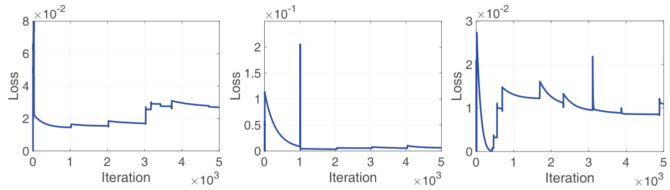
(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 11: Freedom Degree evaluation in multi-round scenario.

is reasonable because when M increases, the solution space is larger than before. Moreover, when N rises, Q_s grows because we can select more users than before. Similarly, we change M from 10 to 50, N from 2 to 10 for Gowalla in Fig. 9b and for Foursquare in Fig. 9c, and get the similar results.

Running time: The computational complexity is a critical metric. When $M = 20$, we change N from 2 to 5 and conduct the simulations of URMB and Optimal strategies for the single-round scenario. As shown in Table I, for all three datasets, when N is small, the running time of the Optimal strategy is similar to that of URMB. But when N grows slightly, the running time will rise explosively. Consequently, URMB is efficient to find the optimal solution.

QoD for multi-round scenario: To evaluate the performance on total QoD Q for the multi-round scenario, we conduct extensive simulations with respect to different variables as shown in Fig. 10. At first, when $N = 10$ and $K = 200$, we change M from 30 to 49. The results in Figs. 10a-10c show that Q increases monotonically with M, and URMB has advantages over other strategies. CUCB is the second-best strategy since it considers both exploitation and exploration but ignores the effect of the collaboration likelihood. Meanwhile, the results of Exploitation and Exploration are not good because pure exploration or exploitation is unreasonable. Note that when M increases, Q has some sudden rises because that some users with strong ability are added and selected. Next, when we change N from 10 to 30 while $M = 50, K = 200$, the results are shown in Figs. 10d-10f. We can conclude that Q also rises with N, and URMB always outperforms other strategies. The differences of these strategies are not obvious,



(a) Brightkite (b) Gowalla (c) Foursquare

Fig. 12: Loss evaluation in multiple rounds.

because when N is close to M, the user sets selected by these strategies are similar. When $N = M$. Then we improve the number of rounds K when $M = 50, N = 10$ in Figs. 10g-10i. All strategies rise linearly except URMB because more rounds means more feedbacks. In Figs. 10j-10l, Q shows an upward trend with the increase of M and N.

Freedom degree: In the multi-round scenario, URMB recruits the same number of users in each round to perform the task, and then we can get a bounded regret in Theorem 4. Otherwise, the regret is not guaranteed. To prove the robustness of URMB, we define the freedom degree d_f and randomly generate the actual number of selected users N' per round in the range $[N-d_f, N+d_f]$, and conduct simulations as shown in Fig. 11. Specifically, the left y-axis in Fig. 11a represents the total QoD after all rounds, and we find that the changes in total QoD of URMB and Optimal are both slight. The left y-axis represents the change ratio $r_{change} = \frac{|Q_N - Q_{N'}|}{Q_N}$, where Q_N and $Q_{N'}$ represent the total QoD when selecting N and N' users per round respectively. And the values of change ratio are small. The similar results are shown in Figs. 11b-11c. In summary, in the actual multi-round recruitment process, even if the number of selected users N changes dynamically in each round, URMB can still achieve a good performance.

Loss: In the simulations, we record the values of the loss function J in Eq. 19 as shown in Fig. 12, which is actually the difference between $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha_r}$. The x-axis represents the number of iterations in BGD, and the y-axis represents the loss. We find that the value of J often rises suddenly and then gradually decreases. The sudden increase of J indicates that the new observed feedback is added after a round, which makes the difference between expected QoD and observed QoD larger. Then the value of J decreases because the value of $\boldsymbol{\alpha}$ is approaching that of $\boldsymbol{\alpha_r}$ through the update process, which indicates that our update strategy is effective.

VII. Conclusion

In this paper, we argue that when selecting a group of users to perform a cooperative task, we should consider not only a user's objective ability but also its collaboration likelihood with others. In the single-round scenario, we convert the recruitment problem into the min-cut problem and propose an algorithm based on graph theory to find the optimal solution. Furthermore, in the multi-round scenario, we propose the multi-round user recruitment strategy based on the combinatorial multi-armed bandit model (URMB) to balance the trade-off between exploration and exploitation, and prove that it achieves a tight regret bound. Finally, we conduct extensive simulations based on three real-world datasets, and the results show that URMB always outperforms other strategies.

References

[1] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *Communications Magazine IEEE*, vol. 48, no. 9, pp. 140–150, 2010.

[2] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2018.

[3] R. Zhou, Z. Li, and C. Wu, "A truthful online mechanism for location-aware tasks in mobile crowd sensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 8, pp. 1737–1749, 2018.

[4] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, and C. Chen, "Mosaic: A low-cost mobile sensing system for urban air quality monitoring," in *IEEE INFOCOM 2016*, pp. 1–9.

[5] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing," in *ACM MobiSys 2012*, 2012, pp. 379–392.

[6] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *ACM SenSys 2009*, pp. 85–98.

[7] Y. Yang, W. Liu, E. Wang, and J. Wu, "A prediction-based user selection framework for heterogeneous mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2460–2473, 2019.

[8] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *IEEE INFOCOM 2019*, pp. 2053–2061.

[9] A. V. Goldberg, "Finding a maximum density subgraph," EECS Department, University of California, Berkeley, Tech. Rep. UCB/CSD-84-171, 1984.

[10] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

[11] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *JMLR ICML 2013*, pp. 151–159.

[12] C. T. Chou, N. Bulusu, S. Kanhere *et al.*, "Sensing data market," in *Proceedings of Poster Papers of IEEE/ACM DCOSS*, 2007.

[13] Y. Wu, F. Li, L. Ma, Y. Xie, T. Li, and Y. Wang, "A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7648–7658, 2019.

[14] G. Gao, J. Wu, M. Xiao, and G. Chen, "Combinatorial multi-armed bandit based unknown worker recruitment in heterogeneous crowdsensing," in *IEEE INFOCOM*, 2020.

[15] M. Tokic, "Adaptive ϵ-greedy exploration in reinforcement learning based on value differences," in *Springer KI 2010*, pp. 203–210.

[16] R. Agrawal, "Sample mean based index policies by o(log n) regret for the multi-armed bandit problem," *Advances in Applied Probability*, vol. 27, pp. 1054–1078, 12 1995.

[17] A. Hüyük and C. Tekin, "Analysis of thompson sampling for combinatorial multi-armed bandit with probabilistically triggered arms," in *AISTATS*, ser. Proceedings of Machine Learning Research, vol. 89. PMLR, 2019, pp. 1322–1330.

[18] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 307–319, 2018.

[19] E. L. Lawler, *Combinatorial optimization: networks and matroids*. Holt, Rinehart and Winston, New York-Montreal, Que.-London, 1976.

[20] M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of ACM*, vol. 44, no. 4, pp. 585–591.

[21] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender, "Learning to rank using gradient descent," in *JMLR ICML 2005*, pp. 89–96.

[22] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *ACM SIGKDD 2011*, pp. 1082–1090.

[23] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. Magnenat-Thalmann, "Time-aware point-of-interest recommendation," in *ACM SIGIR 2013*, pp. 363–372.