2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)

Quality-based User Recruitment in Mobile CrowdSensing

Yu Lin, Fan Wu, Linghe Kong, Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{yulin.cpp, linghe.kong}@sjtu.edu.cn; {fwu, gchen}@cs.sjtu.edu.cn

***Abstract*—Mobile CrowdSensing has played an important role in our daily life. Data quality evaluation and user recruitment are both important problems in CrowdSensing. Recruiting users who will provide high-quality data guarantees the success of a task. In this paper, we will recruit users based on the data quality. First, by exploiting the historical data that users performed on tasks, we use Compressive Sensing(CS) to predict the data quality that a user will achieve on a task which he has never done before. By partitioning the matrix according to the similarity between users, we propose G(grouping)C(Compressive)S(Sensing). Compared with original CS, GCS is more efficient and has higher precision. Then, we use the predicted data quality to guide user recruitment. We consider a general scenario in the real world. For a task, we expect to use as short as possible time to achieve the expected quality. We both consider offline and online scenarios, and we design the greedy approximation algorithms Off-QBUR (Offline Quality-Based User Recruitment) with logarithmic approximation ratio and On-QBUR (Online Quality-Based User Recruitment) algorithm with linear approximation ratio respectively. We use a real-world dataset to evaluate the prediction of the data quality, and the experiment result shows that our method is efficient and can predict data quality with high precision.**

***Index Terms*—CrowdSensing, data quality, user recruitment, Compressive Sensing**

I. Introduction

In recent years, based on the rapid development of CrowdSensing, more and more CrowdSensing applications appear in our life, such as personal body indexes recording for health care [1], environment monitoring like noise pollution [2], road and traffic condition [3], and indoor localization [4].

Mobile CrowdSensing, compared with the traditional deploying sensors to collect data, the most difference lies in human participation. Human is more intelligent so that they can easily do something like classifying a figure, going to the right place, etc., which is useful to collect data with high quality. But the behaviors of people are so different, for example, some people put their phone in the pocket when monitoring noise pollution, that brings great uncertainty to data quality. The success of a CrowdSensing application depends on the reliable data. We need to extract useful information from these data, so high-quality data is essential.

To evaluate data quality, existing works mainly directly deal with the collected data. The main ideas come from the truth discovery method in the databases field [5], [6], [7], [8]. They usually iteratively compute data quality and data truth, by fixing one of them to calculate the other, then repeat iterations until coverage.

However, in the scenario of user recruitment, we need to recruit users before collecting data. Now, what we should do is to predict the data quality a user may achieve before recruiting him or her. Recruiting right users who may contribute high-quality data is very helpful to the success of the task. It can reduce the data noise, and sometimes reduce the number of users recruited, thus reducing the budget. So, accurately predicting the data quality, recruiting the right users is our main work in this paper.

Here, we call a CrowdSensing application as a task in this paper. There will be many tasks on a platform and the platform recorded data quality that users performed on tasks as historical data. The historical data can be represented as a matrix shown in Figure 1. To predict missing data, intuitively, if a user behaves very closely with some other users, we can speculate on his performance by referring to these similar users. For example, in the matrix in Figure 1, user2 and user3 performed similarly on task 2,3 and 5, then we can predict user2 can perform well on task1 for user3 got 1.0 data quality on it. It also applies to similar tasks. Motivated by this observation, we design an experiment to prove the matrix is redundant. So we can use Compressive Sensing(CS) [9], [10] to predict the missing values.

However, there still have great challenges. The severe lack of historical data is the primary challenge in our work. As shown in our evaluation in section V, we have only 0.024% valid values of the whole matrix. It makes a great influence on the precision and efficiency of matrix completion.

To solve the challenge above, we study the similarity between users to group similar users. As similar users may have more similar quality, they can provide each other with a higher reference value. So, we can partition the original matrix into several sub-matrixes according to the grouping of users. Moreover, after filtering the invalid rows and columns(which are all missing values) in sub-matrixes, we can get some smaller sub-matrixes but with the higher ratio. It can improve the precision and the efficiency by using CS on these sub-matrixes. We call this method G(grouping)CS.

User recruitment is another important problem. There are many criterions to pick users in different scenarios. For example, in [11], they recruited users with most information under budget. In [12], they maximized the user' utilities through reverse combinatorial auction.

In our paper, we consider a common scenario in the real world. That is, we often need data for a specific period, and a user often has its own prefered period. For example, we want to investigate the noise pollution in a residential area, and we may only need to monitor the noise level during 9 p.m. to 7 a.m. next day, as this period is the break time which people care more. For a user, (s)he prefers to the different period, maybe some people do not want to stay up late or get up

978-1-7281-0548-2/18/$31.00 ©2018 IEEE
DOI 10.1109/MSN.2018.00019

early. In this scenario, we should also satisfy the task quality requirement, which is introduced in Section II, so we need to recruit suitable users according to their predicted data quality and their period preference. Moreover, we want to minimize the total user time cost to complete the task, which is helpful to save social time. This time-based scenario is first considered in our work.

Besides, we both consider the online and offline scenarios. According to our offline user recruitment approximation algorithm Off-QBUR and online version algorithm On-QBUR, we get logarithmic and linear approximation ratio performance respectively.

The main contributions of this paper are as follows:

- We study the sparse structure and low-rank property of the data quality matrix, and then we use the compressive sensing to predict data quality.
- We study the similarity between users and use KMeans to partition similar users to the same group. By partitioning original quality matrix into several sub-matrixes according to the grouping of users, we propose G(Grouping)C(Compressive)S(Sensing) to predict missing values and get higher precision and efficiency compared with original CS.
- We use predicted data quality to guide user recruitment in a common scenario called time-based scenario, which has not been investigated before. We both consider online and offline situations, and we propose offline greedy algorithm Off-QBUR with logarithmic approximation ratio and online version algorithm On-QBUR with linear approximation ratio.
- We design experiments for the quality prediction on a real-world CrowdSensing data set, and the results show that our method GCS is efficient and has a high precision.

The rest of the paper is organized as follows. Section II models the problems. Section III introduces data quality estimation. Section IV introduces quality-based user recruitment in both offline and online scenarios. Evaluation will be shown in section V. We will review related work in section VI, and section VII concludes the paper and presents our future work.

II. Problem Formulation

In this section, we present our system in detail. First, we give an overview of our CrowdSensing system. Then, we introduce data quality estimation model and user recruitment model respectively.

A. System Overview

As Figure 1 shows, in our mobile CrowdSensing system, there are five major components: Task publishing platform, a set of mobile users, a set of customers, quality estimation model and user recruitment model.

In this system, customers publish a task on the task publishing platform, they specify the period, location, target monitoring object and data quality requirement of the task, then task publishing platform pushes the task to mobile users. Mobile users who want to participate in this task submit their prefered period D(in our time-based scenario) to collect data, then Quality Estimation Model predicts data quality that these users will achieve. After that, User Recruitment Model recruits a subset of these users to collect data based on predicted data

Fig. 1. system overview

quality and prefered D. After users return data to the platform, the platform determines their true data quality by some methods(manual inspection or other automated methods [13], [14]). These data quality records are added to historical data.

B. Quality Estimation Model

In a task publishing platform, there already have some historical data. The data quality is a value from 0 to 1. So we get a matrix shown in Figure 1, called quality matrix(QM). In QM, there are many missing values, and our target is to predict them so that we can predict the data quality that a user will perform on a task which (s)he has never done before.

Intuitively, the main factors that affect the data quality most are the behavior of users and the performance of their mobile sensors. If some people have similar behavior and mobile sensors with similar performance, they may get similar data quality on the same task. Based on this motivation, we design an experiment to prove that the data in QM is redundant.

We extract 19×7, 135×6, 11×8 three full quality matrixes X from our dataset. The problem of lack of historical data is severe in the real world, and that is why we can only extract such these small matrixes.

Based on the Singular Value Decomposition(SVD), we can get the rank of a matrix X. Figure 2 shows the singular values of our three test quality matrixes. As the scale of these three matrixes is different, so we normalize the axis to 1. The result shows that the top several singular values contribute the most of energy in X. We find that the top 20% singular values contribute 80% of the sum of all singular values, so we can use these top 20% singular values to approximate the matrix X. Now we get the low-rank property of Quality Matrix, and it is the prerequisite for using Compressive Sensing(CS).

CS is a generic data reconstruction method to recover whole data with just a few sampled data. CS requires that QM be sparse or have the low-rank structure which shows that QM is redundant. Moreover, it is evident that QM is sparse as so many missing values in QM.

Now, we formulate our quality prediction model. We have some definitions for ease of statement.

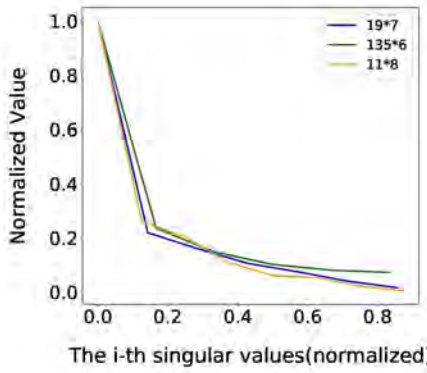


Fig. 2. Low-Rank structure Discovery of Quality Matrix

Definition 1 Full Quality Matrix(FQM). FQM is the real data quality matrix which is unknown. For a platform with n users and m tasks, its FQM is denoted as F_{n*m}, where each entry F_{ij} denotes the real data quality of user i performs task j.

Definition 2 Binary Matrix(BM). B denotes BM, and B_{ij} indicates that if F_{ij} in FQM is missing:

$$B_{ij} = \begin{cases} NaN & \text{if } F_{ij} \text{ is missing} \\ 1 & \text{otherwise} \end{cases}$$

where **NaN** means Not a Number.

Definition 3 Quality Matrix(QM). QM records the history data quality of users performed on tasks. QM is denoted by X, where X_{ij} is the data quality user i performed on task j. Due to QM has missing data, the element of X_{ij} is either F_{ij} or NaN. So X can be denoted by:

$$\mathrm{X} = \mathrm{F} \circ \mathrm{B}$$

where $\circ$ denotes the element-wise product of two matrixes.

Definition 4 Quality Matrix Reconstruction Algorithm. Reconstruction Algorithm CS reconstructs a full quality matrix $\hat{F}$ from quality matrix X, so that:

$$CS(X_{n*m}) = \hat{F}_{n*m} \approx F_{n*m}$$

Definition 5 Error Ratio(ER). ER is a metric for measuring the error after reconstruction [15]:

$$\epsilon = \frac{\sum_{i,j:B_{ij}=NaN} |F_{ij} - \hat{F}_{ij}|}{\sum_{i,j:B_{ij}=NaN} 1}$$

where B_{ij} = NaN indicates that only errors on the missing data are considered.

In our work, given a quality matrix X, we use CS to reconstruct it to $\hat{F}$, so that $F \approx \hat{F} = CS(X)$. And ER is used to measure the performance.

C. User Recruitment Model

As we mentioned before, a task has a period T = $\{t_1, t_2, ..., t_m\}$ with m time slots and a quality requirement θ. There are a set of users U = $\{u_1, u_2, ..., u_n\}$, and we have already predicted the data quality Q = $\{q_1, q_2, ...q_n\}$ they may achieve. Our target is to select a subset of U to take this task and satisfy the quality requirement.

First, the quality requirement θ means the quality of each slot should be at least θ. For each slot, we may have several users collecting data with different data quality, so we should find a method to measure the quality of Crowd(QoC). The QoC function should satisfy the law of diminishing marginal utility [16]. Here, the increment of the QoC is the marginal utility, and as data quality grows, the marginal utility is lower. [17] gives several Quality of Crowd(QoC) models. Here, we use the Hyperbolic tangent model in our scenario. We should note that any other functions satisfying property said above are also suitable for our problem. Using a general one function will be in our future work.

Second, we want to complete a task as short time as possible. We minimize the sum of time cost of the recruited users and satisfy the quality requirement θ in each slot. It is meaningful as we could decrease the total social time cost.

Third, we consider both online and offline scenarios. Moreover, we assume that there are enough users to satisfy the quality requirement in each slot. It is reasonable because in practice we can give more money to attract more users or only decrease the expected quality requirement.

In the offline scenario, we know period D of all users. In summary, we can formulate the offline version as an optimization problem:

$$\begin{aligned} & min(\sum x_i |D_i|) \\ s.t. \quad & tanh(\sum_{i:j \in D_i}^{n} q_i x_i) \geq \theta, \ 1 \leq j \leq m \\ & 0 \leq q_i \leq 1 \\ & x_i \in \{0, 1\} \end{aligned} \tag{1}$$

where n is the number of users, m is the number of time slots, and D_i is the preferred period of user i.

In the online scenario, we can get information of all users whose preferred period D contain slot t_i only until t_i comes. That the arrival time of users is arbitrary brings new challenge. Shown in Figure 3, if we know all users' information from the beginning, then we can easily get the optimal selection, that is we recruit users 1, 2, 4 as $tanh(0.3 + 0.5) > \theta = 0.6$ and $tanh(0.9) > \theta$, which cost minimum time of 4. However, in the online scenario, u_4 is unknown until slot 2 comes, so now we do not know if we should recruit user 3 in slot 1.

III. Data Quality Estimation

In this section, we first use Compressive Sensing to reconstruct the full quality Matrix by matrix complete method, then we study the similarity between users and use the KMeans method to partition them into different groups and propose GCS.

A. Compressive Sensing

In this section, given a sparse Quality Matrix X, Compressive Sensing reconstruct the full Quality Matrix $\hat{F}$. Matrix completion technic [18], [19], [20] can be used to solve this problem by exploiting low-rank property:

$$\begin{aligned} & min \ rank(\hat{F}) \\ & s.t. \ X = B \circ \hat{F} \end{aligned} \tag{2}$$

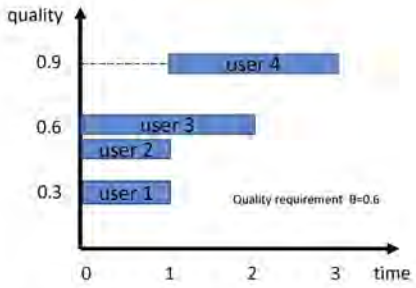


Fig. 3. Online User Recuitment. Quality requirement is 0.6, and user 1-4 has quality 0.3, 0.4, 0.6, 0.9 respectively.

Equation (5) is an NP-hard problem. And we observe that the convex hull of rank($\hat{F}$) on the set $\{\hat{F} \in \mathbb{R}^{n\times m} : \|\hat{F}\|_F^2\}$ is the nuclear norm of $\hat{F}$. The nuclear norm is equal to the sum of the singular values of the matrix. So based on the Singular Value Decomposition(SVD), Let

$$\|\hat{F}\|_* = \sum_{i}^{min(n,m)} \sigma_i(\hat{F}) \tag{3}$$

Then we can get the approximate solution by minimizing the nuclear norm of $\hat{F}$, so we can convert Equation (5) to:

$$\begin{aligned} &min\ \|\hat{F}\|_* \\ &s.t.\ X = B \circ \hat{F} \end{aligned} \tag{4}$$

Equation (7) is a convex optimization problem and we can get the solution through Semi-Definite Programming(SDP) [21].

B. Grouping Compressive Sensing

In GCS, we need to group similar users. First, we should characterize the feature vector of a user. For a user u_i, he has a quality vector r_i which is the i-th row of the quality matrix. Most of the users just do a little number of tasks, so there are so many missing values in r_i. We should not use r_i to study similarity between users.

As we discussed in section II, one of the critical factors influencing data quality is mobile sensors. Sensor type is a good standard for differentiating task. So, intuitively, we can construct a new quality vector v_i for each user, where each element in v_i is the average quality that user i performed on a particular kind task.

Second, We use cosine distance to characterize the similarity between users. Cosine distance characterizes the angle between two vectors. In a matrix, if two row vectors are parallel, their cosine distance is 0, and these two row vectors are redundant. It is helpful to the matrix completion.

We use a clustering method to partition users, and here we use KMeans [22] method where parameter K is tunable. After we partition users, we further partition the original quality matrix into several sub-matrixes. For each sub-matrix, we can further remove the invalid column whose all values are missing and then apply CS to recover the rest of missing values.

IV. Quality-Based User Recruitment Algorithm

In this section, we will introduce our greedy algorithms in offline and online scenarios respectively.

A. Offline User Recruitment

We first convert Equation (3) to a standard form. As $tanh(x)$ is a bijection function, so if we have θ, for each slot j, let $\sum_{j\in D_i} q_i x_i \geq tanh^{-1}(\theta)$, then we can get $tanh(\sum_{j\in D_i} q_i x_i) \geq \theta$. So we can convert Equation (3) to:

$$\begin{aligned} &min \sum x_i c_i \\ &s.t.\quad A\vec{x} \geq \vec{b} \\ &\quad\ x_i \in \{0,1\} \end{aligned} \tag{5}$$

where A is a $m*n$ matrix where a_{ij} represent data quality of user j in time slot i, $a_{ij} = q_j$ if $i \in D_j$ else be 0. $\vec{b}$ is an m-dim vector where each element is $tanh^{-1}(\theta)$. It is a covering-like problem. In [23], Dobson analyzed the error bound of his greedy heuristics when the nonzero elements of A and b are at least 1. Here, we can easily convert A and $\vec{b}$ satisfy it in our scenario. In our scenario, we always set the minimum quality restriction δ as poor quality data does not make any sense to us. We can divide each element in A and $\vec{b}$ by δ so that each nonzero element is at least 1. So we get offline Quality-Based User Recruitment Algorithm(Off-QUBR). After we convert A and $\vec{b}$ to be more than 1, then the following procedure is the same with the second greedy algorithm in [23].

Combine the analysis in [23], let C_1 be the result of time cost according to Off-QBUR, we can get the error bound:

Here, d_j is the number of nonzero elements in column j, $H(d)$ is the first d terms of the harmonic series: $H(d) = \sum_{i=1}^{d} 1/i$, and C_* is the optimal minimum time cost.

In our scenario, $0 \leq a_{ij} \leq 1$ and $d_j \leq T$, then we can get the sum of each column is at most T, so we can further get the approximation ratio:

$$\begin{aligned} \frac{C_1}{C_*} &\leq max_{1\leq j\leq n}\{log(\sum_{i=1}^{T} a_{ij}/\alpha) + 1 + H(d_j)\} \\ &\leq log(T) + 1 + H(T) - log(\alpha) \end{aligned}$$

$H(d)$ is also logarithmic. So the approximation ratio of Off-QBUR is logarithmic.

B. Online User Recruitment

We first consider an easy case. That is, when we recruit a user u_i, we can delegate him to collect data in just part of time slots in D_i but not must be the whole D_i. We call it a part-time situation(PS).

We can easily get the optimal solution for PS. For time slot t_i, we recruit several the highest quality user to collect data on t_i until quality requirement is satisfied. It is obvious that the cost time for each time slot is minimum, so it is the optimal solution. Let this cost be C_p, and recruited user set be S_p. C_p is the lowest bound of any feasible solution.

Motivated by PS, we design our On-QBUR algorithm. In our On-QBUR algorithm, when time slot t_i comes, there are some users in the system whose D contain t_i. Same with PS's strategy, we recruit users based on the data quality from high to low until QoC satisfy the quality requirement on t_i. Users recruited need to collect data in his D_i. Algorithm 1 shows the process of t_i time slot.

After m round, we can get the final result. Let the total cost time be C_f and recruited user set be S_f. Now we can

analyze the approximation ratio of algorithm 1. Here we can easily get $S_f \subseteq S_p$ and $C_p \leq C_f$. For $S_f \subseteq S_p$, in each slot t_i, we both recruit the highest quality users from high to low, and in PS, a(like a in algorithm 2) begins with 0, but not in On-QBUR($a = \sum^{S_k|t_i \in D_{S_k}} Q_{S_k}$), so On-QBUR can satisfy the quality requirement earlier. It means that the users recruited in On-QBUR is a subset of that in PS in each round, so $S_f \subseteq S_p$. It is evident that $C_p \leq C_f$ as C_p is the minimum time cost.

For C_p, we assign u_i to collect data in the part of D_i. Let $A_i \subseteq D_i$ be the time slots we choose. Then we can get:

$$\begin{aligned} C_f &= \sum_{u_i \in S_f} |D_i|/|A_i| * |A_i| \\ &\leq \sum_{u_i \in S_p} max_{u_j \in S_f}\{|D_j|/|A_i|\} * |A_i| \end{aligned} \tag{6}$$

And $max_{u_j \in S_p}\{|D_j|/|A_j|\} \leq |T|$ as $|D_j|$ is at most $|T|$ and $|A_j|$ is at least 1. So,

$$C_f \leq |T| C_p \tag{7}$$

Let C_* be the optimal solution for online scenario, as C_p is the lowest bound, so:

$$C_p \leq C_* \tag{8}$$

then combined with $C_p \leq C_f$, we can get:

$$\frac{C_f}{C_*} \leq \frac{C_p}{C_*} \leq \frac{C_2}{C_p} \leq |T| \tag{9}$$

So we prove that our On-QBUR has a linear approximation ratio.

$$\frac{C_1}{C_*} \leq max_{1 \leq j \leq n}\{log(\sum_{i=1}^{T} a_{ij}/\alpha) + 1 + H(d_j)\} \tag{10}$$

Algorithm 1: Online Quality-Based User Recruitment algorithm

Input:
S: the set of users who have been already recruited
D, Q: users' corresponding period D and quality Q
t_i: the current time slot
U: users arrived whose prefered period contain t_i
θ: quality requirement
Output:
S: updated S

```
X ← {(U_j, D_{U_j}, Q_{U_j})|U_j ∈ U}
X ← Order X descending by quality
a = Σ^{S_k|t_i∈D_{S_k}} Q_{S_k}
j = 1
while tanh(a)< θ do
    S = S ∪ {X_j.U}
    a = a + X_j.q
    j = j + 1
end
return S
```

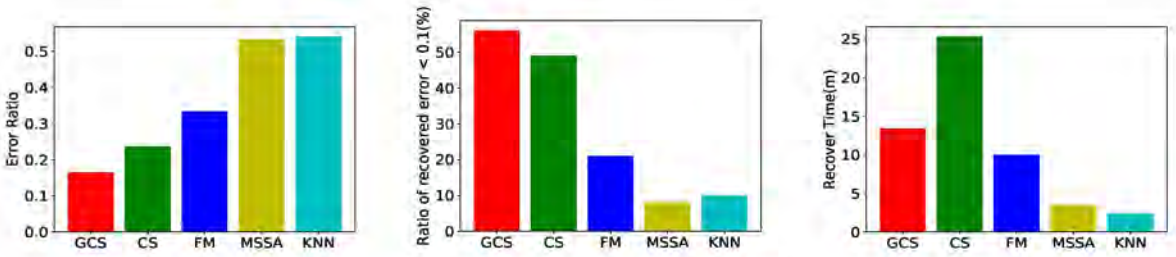
V. Evaluation

In this section, we show our experiments on data quality prediction and user recruitment.

Our first experiment is to evaluate the performance of CS and GCS. We use a real-world dataset coming from a crowdsourcing company. The original dataset contains 51165 users and 10690 tasks and 132145 quality values which is a score between 0 to 1. There are only 0.024% values exist in this data quality matrix. Like [15], we compare our GCS with some interpolation methods. They are compressive sensing (CS), Multi-channel Singular Spectrum Analysis (MSSA) [24], and K-Nearest Neighbor (KNN) [25]. As matrix completion is similar to recommender problem, we also compare our method with FM [26], which is a popular algorithm in the recommender system.

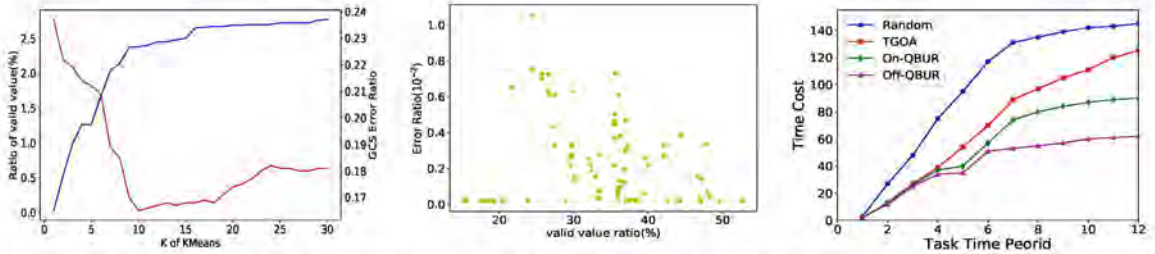
Our experiment procedure is that, for a quality matrix X, X has some valid values. We randomly select 10% of them as test data, and the rest of valid values as training data, then we recovery the whole matrix based on the training data, and we can further get the recovery error on the test data. We average these recovery errors on test data and take it as the prediction performance of the corresponding method. The parameter K in KNN is set to be 16. The parameter M in MSSA is set to 32 as suggested by [24]. For FM, following [26], we construct a sparse vector $\vec{x}$ for each valid value. For a valid value that user i performed on task j, its sparse vector contains three segments, the first segment is user id i which is encoded as one-hot vector, the second segment is the task id j which is also encoded as one-hot vector, and the third segment records valid values that other tasks the user i has performed. So the dimension of $\vec{x}$ will be $51165 + 10690 + 10690 = 72545$, and its target y is the data quality X_{ij}. We use the FM implement LIBFM [27] which provided by authors and choose the MCMC learning method by default. For GCS, we classify task based on sensor type, in fact, there are mainly 6 type sensors are used in this platform, and we set other sensors as 'other' type, then we generate 7-dimension quality vector v_i and use KMeans to group users.

Figure 4(a) shows the comparison results. We can see that the error of MSSA, KNN is nearly random($\approx$ 0.5). Because the prediction of these methods mainly refer to the neighbors' value of the target, but in the quality matrix, neighbors' values can not provide much information. Moreover, the result of FM is also not promising, as FM performs well when the history data is enough, but there are only 0.024% valid data, and it has so many parameters, which makes FM easier to overfit. This result shows that the CS is suitable for our quality prediction problem. For GCS, in Figure 5(a), the red line shows the variation of error ratio with the K value of KMeans, and the blue line shows the ratio of valid value after grouping. As K increases, we can get more sub-matrices, after filtering the invalid column in each sub-matrix, we can increase the ratio of valid value. We can see that at the beginning, as the ratio of valid value increases, the error gradually decreases, this shows that enough data(high valid value ratio) is significant to matrix completion. However, when the K continues to increase, the ratio of valid value tends to be gentle, and the error increases slightly, this is because



(a) Error Ratio (b) Ratio of recovery error less than 0.1(%) (c) Recovery Time

Fig. 4. Results of Quality Matrix reconstruction



(a) Distribution of Error Ratio and Ratio of valid value over K value (b) Distribution of Error Ratio over Ratio of valid value (c) Time cost to complete task of different algorithms

Fig. 5. Grouping result and User Recruitment result

when K continues to increase, some of the relevant users are separated, causing some useful information to be lost. GCS get the best error ratio 0.165 when $K = 10$. We further explain the relationship between the error ratio and the ratio of valid value in Figure 5(b). In the process of grouping(K from 1 to 30), we get many relatively small sub-matrices but with a high ratio($\geq 15\%$) of valid value. These sub-matrices have a very low error ratio(≤ 0.01). Figure 5(b) shows that the error ratio is positively related to the valid value ratio, and high ratio means a higher probability of getting a lower error ratio.

We further compare the performance between CS and GCS. In Figure 5(a), we see that the error ratio of CS is 0.232. Compared to CS, we can see that GCS reduced errors about 0.067 and have about 28.4% precision performance improvement. Figure 4(b) shows the precision performance improvement in another aspect, we statistics the ratio of recovery values whose recovered error are less than 0.1, we can see that there have about 3.9% performance improvement, although it is not apparent. In Figure 4(c), we compare the efficiency between CS and GCS. We record the average time cost of reconstruction process in CS and GCS, and we can see that GCS has 11.4% efficiency performance improvement compared to CS. FM, MSSA and KNN are efficient but low precision.

In our second experiment, we compare the time cost to complete task returned by our offline user recruitment algorithm, online user recruitment algorithm, a random greedy algorithm, and a existing online mobile Micro-Task allocation algorithm TGOA [28] in Crowdsourcing.

The experiment procedure is: We set that 100 users (make sure there are enough users to complete the task) want to participate in the task and select them from all of the users at random. We set the minimum quality $\delta = 0.2$ and the quality requirement $\theta = 0.9$, then we test T from 1 to 12. For each T, we generate users' period D randomly according to T, and users' data quality is calculated by GCS. We run these four algorithms and get the results. Of course, we repeat the experiment 10 times to get the average performance for each T. Besides, for the random greedy algorithm, it chooses a user randomly until satisfying the quality requirement.

In [28], they studied an online mobile Micro-Task allocation problem and proposed TGOA with $1/4$-competitive ratio under the online random order model which means the arrival of users is random.We compare it with our method by making TGOA adaptive for our scenario. Figure 5(c) shows the results. We can see that when T is small, the results of the four algorithms is close as a small number of users is needed to satisfy the quality requirement. As T grows up, the difference becomes large. We have enough users to participate in, so the Off-QBUR gets better performance than On-QBUR as T grows up. Compare On-QBUR and TGOA, we can see that On-QBUR prefers the user who has higher data quality, and TGOA prefers the user who has higher utility. So TGOA will recruit some users who have not rather high quality but long period, which increase the time cost and this problem is prominent when T grows up. So, in this experiment, when T is small(≤ 6), TGOA, Off-QBUR and On-QBUR have close performance, but when T becomes larger(≥ 7), On-QBUR will reduce about 28.4% time cost compare to TGOA, and Off-QBUR will reduce about 44.8% time cost compare to On-QBUR.

VI. Related Work

Data Quality Estimation: Evaluation of data quality has been widely studied in databases field [5], [6], [7], [8], they mainly iteratively compute data quality and data truth. In mobile CrowdSensing, based on the idea from truth discovery, Peng [13] used EM algorithm to calculate the noise truth and data quality iteratively. Combined with GMM, Liu [14] also used EM algorithm to calculate the data truth and quality level. They all evaluate data quality based on the collected data. To the best of our knowledge, we are the first work to predict data quality from users' historical data in CrowdSensing.

Compressive Sensing: Compressive Sensing [18], [9] is a powerful tool to reconstruct a sparse matrix based on the sparsity or low-rank property of matrix. A large number of applications based on CS have appeared, such as network traffic reconstruction [29], environmental data recovery [15] and face recognition on smartphones [30]. Such Matrix completion problem is also commonly investigated in the recommender system, and FM [26] is a popular method. With the development of machine learning, some improved algorithm for FM were proposed, they are FFM [31] and DeepFM [32]. However, these methods perform well only when the training data is enough as they should fit a lot of parameters.

Compared to their works, our challenge is the great lack of matrix data. We study the correlation between users, by partitioning more relevant users into the same group, the original matrix is decomposed into several smaller sub-matrixes. We call it as GCS that is proved to have higher precision and efficiency.

Approximation Algorithm of Set Covering Problems: Set Covering problem is widely studied in mathematics. Johnson [33] proved in the mid 70's that a simple greedy heuristic achieves a ratio of $0.7log_2N$, where $N = |U|$ is the number of different elements. Chvatal [34] generalized it to be a weighted case. The result was further generalized in different directions, for example, to the Integer Programming problem of minimization with nonnegative coefficients [23], to the case of submodular constraints [35], and to a continuous version [36].

VII. Conclusion

In this paper, we discover the low-rank structure of Quality Matrix, after partition Quality Matrix into several submatrixes, we propose a novel approach GCS to predict data quality. Our experiment results show that GCS has the great performance in predicting data quality. Then we use predicted data quality to guide user recruitment. We consider both offline and online scenarios and design Off-QBUR greedy algorithm with logarithmic approximation ratio for the offline scenario and On-QBUR greedy algorithm with linear approximation ratio for the online situation.

Now, we introduce our future work. First, we will investigate more methods to solve great lacking historical data problem. Second, to characterize the QoC, every function satisfied the law of diminishing marginal utility can be used not only the Hyperbolic tangent model, so more general one will be researched. In the last, based on the expect quality estimation error, we can take more control on user recruitment to satisfy quality requirement and minimize user time cost. That is, give a convince probability to the user recruitment results according to the quality estimation error.

References

[1] R. Pryss, M. Reichert, and B. Langguth, "Mobile crowd sensing services for tinnitus assessment, therapy, and research," in *MS*, 2015.

[2] N. Maisonneuve and Stevens, "Noisetube: Measuring and mapping noise pollution with mobile phones," *ITEE*, 2009.

[3] Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu, "Crowdatlas: Self-updating maps for cloud and personal use," in *MobiSys*, 2013.

[4] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *MobiCom*, 2012.

[5] D. M. Strong, Y. W. Lee, and R. Y. Wang, "Data quality in context," *Communications of the ACM*, vol. 40, no. 5, pp. 103–110, 1997.

[6] Q. Li and Y. Li, "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, 2014.

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

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

[9] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, 2006.

[11] S. Yang and F. Wu, "Selecting most informative contributors with unknown costs for budgeted crowdsensing," in *IWQoS*, 2016.

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

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

[14] S. Liu, Z. Zheng, F. Wu, S. Tang, and G. Chen, "Context-aware data quality estimation in mobile crowdsensing," in *INFOCOM*, 2017.

[15] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *INFOCOM*, 2013.

[16] T. Polleit, "What can the law of diminishing marginal utility teach us," *Mises Daily, Feb*, vol. 11, p. 2011, 2011.

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

[18] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, 2009.

[19] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.

[20] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, 2010.

[21] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[22] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *BSMSP*, 1967.

[23] G. Dobson, "Worst-case analysis of greedy heuristics for integer programming with nonnegative data," *MOR*, 1982.

[24] H. Zhu, Y. Zhu, M. Li, and L. M. Ni, "Seer: Metropolitan-scale traffic perception based on lossy sensory data," in *IEEE INFOCOM 2009*, 2009.

[25] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, 1967.

[26] S. Rendle, "Factorization machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, ser. ICDM '10, 2010.

[27] ——, "Factorization machines with libfm," *ACM Trans. Intell. Syst. Technol.*, 2012.

[28] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing." 2016.

[29] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *SIGCOMM*, 2009.

[30] Y. Shen and W. Hu, "Face recognition on smartphones via optimised sparse representation classification," in *IPSN*, 2014.

[31] Y. Juan, Y. Zhuang, W.-S. Chin, and C.-J. Lin, "Field-aware factorization machines for ctr prediction," in *RecSys*, 2016.

[32] Y. Y. Z. L. X. H. Huifeng Guo, Ruiming Tang, "Deepfm: A factorization-machine based neural network for ctr prediction," in *IJCAL*, 2017.

[33] D. S. Johnson, "Approximation algorithms for combinatorial problems," *Journal of computer and system sciences*, 1974.

[34] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of operations research*, vol. 4, no. 3, pp. 233–235, 1979.

[35] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[36] M. L. Fisher and L. A. Wolsey, "On the greedy heuristic for continuous covering and packing problems," *JADM*, 1982.