A Cost-Quality Beneficial Cell Selection Approach for Sparse Mobile Crowdsensing with Diverse Sensing Costs

Zhu, Z.; Chen, B.; Liu, W.; Zhao, Y.; Liu, Z.; Zhao, Z.

DOI
10.1109/JIOT.2020.3024833

Publication date
2021

Document Version
Author accepted manuscript

Published in
IEEE Internet of Things Journal

A Cost-Quality Beneficial Cell Selection Approach for Sparse Mobile Crowdsensing with Diverse Sensing Costs

Zhengqiu Zhu[1], Bin Chen[1,*], Wenbin Liu, Yong Zhao, Zhong Liu, and Zhiming Zhao*, *Senior Member, IEEE*

***Abstract*—The Internet of Things (IoT) and mobile techniques enable real-time sensing for urban computing systems. By recruiting only a small number of users to sense data from selected subareas (namely cells), Sparse Mobile Crowdsensing (MCS) emerges as an effective paradigm to reduce sensing costs for monitoring the overall status of a large-scale area. The current Sparse MCS solutions reduce the sensing subareas (by selecting the most informative cells) based on the assumption that each sample has the same cost, which is not always realistic in real-world, as the cost of sensing in a subarea can be diverse due to many factors, e.g. condition of the device, location, and routing distance. To address this issue, we proposed a new cell selection approach consisting of three steps (information modeling, cost estimation, and cost-quality beneficial cell selection) to further reduce the total costs and improve the task quality. Specifically, we discussed the properties of the optimization goals and modeled the cell selection problem as a solvable bi-objective optimization problem under certain assumptions and approximation. Then, we presented two selection strategies, i.e. Pareto Optimization Selection (POS) and Generalized Cost-Benefit Greedy (GCB-GREEDY) Selection along with our proposed cell selection algorithm. Finally, the superiority of our cell selection approach is assessed through four real-life urban monitoring datasets (Parking, Flow, Traffic, and Humidity) and three cost maps (i.i.d with dynamic cost map, monotonic with dynamic cost map and spatial correlated cost map). Results show that our proposed selection strategies POS and GCB-GREEDY can save up to 15.2% and 15.02% sample costs and reduce the inference errors to a maximum of 16.8% (15.5%) compared to the baseline-Query by Committee (QBC) in a sensing cycle. The findings show important implications in Sparse Mobile Crowdsensing for urban context properties.**

***Index Terms*—Sparse Mobile Crowdsensing, Cost-quality beneficial cell selection, Cost inconstancy, Bi-objective optimization.**

I. Introduction

THE rapid development of the Internet of Things (IoT) and mobile computing technologies [1], [2] promotes the emergence of intelligent, open, and large-scale sensing mechanisms, which allow citizens to effectively collect and share real-time information, and enable innovative urban computing solutions to tackle city-level challenges like carbon emission [3], noise [4], traffic congestion [5], and infrastructure status [6]. With the widely adopted sensor-rich smartphones, mobile crowdsensing (MCS) [7], [8] plays an increasingly important role in urban computing for addressing various urban-scale monitoring needs. To ensure high-quality sensing services, MCS systems often require a large number of mobile users to satisfy the high coverage ratio (quality metric) [9], [10], [11], which is often expensive and unrealistic when budgets and the number of participants are limited. Since sensing maps usually have a low-rank feature, researchers proposed to use compressive sensing (CS) [12] techniques to collect data from only a few subareas, and then to deduce the missing data of unsensed cells by exploiting the inherent correlation of sensing data. In this way, Wang et al. [13] proposed Sparse MCS to reduce the number of required samples but still keep a predefined data quality required for the MCS organizers.

In Sparse MCS, one important issue is cell selection; the organizer needs to decide where and when to collect sensed data from mobile users. Data of different MCS systems may involve diverse spatiotemporal correlations, it is thus a non-trivial task to design proper cell selection strategies. We reviewed the following methods used by existing Sparse MCS studies. The first one is based on Query by Committee (QBC), which selects the next salient cell to sense by calculating the uncertainty of the missing data in unsensed cells. While QBC only considers the subarea which is the most uncertain at that moment [14], Liu et al. [15], [16] proposed a deep Q-network based cell selection strategy, which can approximate the global optimal strategy relying on sufficient data training. Different from the above-mentioned approaches, Xie et al. [17] proposed a bipartite-graph-based sensing scheduling scheme to actively determine the sample locations, which is suitable for linear systems and requires the knowledge of matrix rank in advance. Thus, it is hard for this method to apply in real-world nonlinear systems. More importantly, these methods only aim to maximize the informativeness, but without considering the diversity of sample costs when selecting subareas for recovering the unsensed values.

To the best of our knowledge, the state-of-the-art Sparse MCS techniques assume that the sensing cost is constant spatially and temporally. However, in practical MCS activities, an activity organizer has to consider the diversity of sensing

Zhengqiu Zhu, Bin Chen, Yong Zhao, and Zhong Liu are with the College of Systems Engineering, National University of Defense Technology, Changsha, Hunan 410073, China. (e-mail: zhuzhengqiu12@nudt.edu.cn; chenbin06@nudt.edu.cn; zhaoyong15@nudt.edu.cn; phillipliu@263.net).

Wenbin Liu is with the College of Computer Science and Technology, Jilin University, Changchun, China. (e-mail: liuwb16@mails.jlu.edu.cn).

Zhiming Zhao and Zhengqiu Zhu are with the research group of Multi-scale Networked Systems at University of Amsterdam (UvA), Netherlands. (e-mail: z.zhao@uva.nl)

costs for several reasons: (1) Sensors possessed by mobile users are inherently diverse, and measurement accuracy largely depends on the sensors. Generally, data reports with high precision should be offered with higher rewards; (2) The cost of reporting a sensed data to the organizer varies based on the network condition, distance to the nearest cell tower, cellular data plan, or other concurrent activities on the device [18]; (3) Prior work also found evidence that the final cost may also be affected by the subjective perception of participants. For instance, a user would ask for a higher reward when he is running out of battery [18]. In brief, different conditions can result in vastly different costs in crowdsensing activities. Therefore, this paper aims to improve the existing MCS solutions by reducing sample costs and inference errors (i.e. the quality metric for Sparse MCS systems) with explicit incorporation of cost-diversity into cell selection strategies.

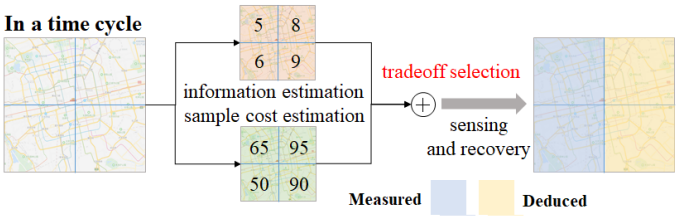


Fig. 1. Example case

To realize the research target of this paper, an effective cell selection strategy is crucial. Since a cell with low-cost might run counter to the need of collecting more information for inferring the missing data. For instance, if we have quantified the informativeness and sample cost in all subareas (shown in Fig. 1), one naïve approach is always to select sample locations with the lowest cost, which will inevitably result in poor recovery of missing data. In other words, the predefined data quality requirement cannot be satisfied. Another naïve approach is to simply divide the informativeness by the sample cost. But it may fail when one of the two factors dominates the other. Suppose that we have an elaborate selection strategy selecting more informative subareas on the premise of reducing or not increasing the overall cost. Since more cells are sensed with real values, the inference error is also improved. For instance, if we set the cost budget in one selection to 120 (e.g. the unit of cost is CNY), a selection containing the top-left cell and the bottom-left cell in Fig. 1 is evidently better than the selection of the bottom-right cell because more information is obtained. However, such a selection mechanism is difficult to design. First, a quantitative model is needed to model information on selected subareas for inferring the missing data. Second, a proper sample cost estimation method is required for accurate estimations of sample cost. Finally, proper strategies to find cells that are low-cost, and yet collect sufficient information are the most important requirement for capturing the underlying data structure to enable accurate recovery.

To tackle the above-mentioned challenges, this paper thus aims to contribute:

(1) We select four city-level datasets of various application domains and verify the inherent low-rank feature and spatial-temporal correlation feature in these urban datasets, which is the basis of the research in this paper.

(2) To the best of our knowledge, we are the first work providing comprehensive modeling of the cost diversity in Sparse MCS and considering dynamic cost budget in bi-objective cell selection problem. Incorporating such a cost enhances the practicability of the Sparse Mobile Crowdsensing methods. A novel cell selection approach, consists of three steps: information modeling, cost estimation and cost-quality beneficial selection, is proposed in this paper. Significantly, three important cost factors, i.e. routing cost, measurement cost, and perception cost, are discussed in detail and proposed to formulate a cost estimation function. Finally, the cell selection process is formulated as a bi-objective optimization problem with the target of maximizing the informativeness in the selected subareas for recovery and minimizing the total sample costs. To solve the optimization, we propose two selection strategy, namely Generalized Cost-Benefit Greedy (GCB-GREEDY) selection and Pareto Optimization Selection (POS).

(3) We provide extensive discussions on the objective functions of our cell selection optimization problem. Note that submodularity is an attractive property encoding a natural diminishing returns condition. But submodularity and monotonicity are not always acquired in any conditions. Since a subset selection problem is NP-hard, our cell selection problem is definitely NP-hard and only solvable under certain assumptions (sufficient participants in each subarea and thus routing costs can be ignored). Under the solvable condition, we formalize the cost-quality beneficial cell selection algorithm for a MCS task.

(4) We discuss the potential of applying the cost-quality beneficial Sparse MCS approach on urban context computing and activation. By integrating the power of crowds, the urban context can be sensed in a cost-aware and sparse way over a large-scale region; By leveraging the wisdom of crowds, the efficiency of smart city systems is optimized, like the re-balancing problem of shared bikes in modern cities.

(5) We evaluate the performance of our proposed cell selection strategies using real-world cases. After taking QBC as the information modeling method and generating three types of cost maps (i.i.d with dynamic cost map, monotonic with dynamic cost map and spatial correlated cost map) based on the cost estimation function, we conduct extensive experiments on Parking, Flow, Traffic, and Humidity monitoring datasets. The results can verify the effectiveness and feasibility of our proposed approach and strategies. Taking humidity sensing as an example, our proposed strategies outperform the baselines QBC and SIMP-GREEDY by lowering up to 15.2% and 8.5% sample costs while reducing inference errors to a maximum of 10.1% (3.8%). Similar tendencies are observed in the other three sensing tasks.

The remainder of the paper is organized as follows. We first review the related works in Section II. Then, the research problem is formulated and the system model is explained in Section III. Next, our three-step cell selection approach is presented in Section IV. Subsequently, we analyze the potential of cost-quality aware Sparse MCS assisted urban sensing

and actuation. Finally, we evaluate the performance of the proposed strategies in Section VI and conclude this paper in Section VII.

II. Related works

In this section, we review the related work from three perspectives: (1) Multi-objective task assignment methods in MCS, (2) Sparse MCS, and (3) Techniques in Subset selection.

A. Multi-objective task assignment methods in MCS

Quality, coverage, cost, and etc. are main factors affecting the performance of designed algorithms in task allocation process of MCS. Therefore, the recent researches aimed to formulate the task allocation problem from a multi-objective perspective considering the aforementioned factors. In [19], Liu et al. discussed the existing strategies to reduce the resource cost and improve quality of service (QoS). Specifically, Xu et al. [20] proposed a Compressive Crowdsensing (CCS) framework to realize reduced amounts of collected data and acceptable levels of overall accuracy at the same time. The limitation of this work is the assumption that the structure and relationships within the data and phenomenon are unchanged from what is observed in historical data. To balance between signal quality and crowdsourcing cost, He et al. [21] proposed an incentive mechanism based on Bayesian Compressive Crowdsensing. The contribution of work is the link between missing value inference and confidence estimation & stopping. Differently, Meng et al. [22] focused on the unevenly distributed user observations over the monitored entities and they designed an integrated framework to realize truth discovery from redundant and sparse data. Xia et al. [23] designed a mobile edge computing architecture to select the minimal set of users in each time cycle with maximized user spatiotemporal coverage while keeping the predefined data requirement. To provide a unified task assignment design, UniTask [24] optimized the overall system utility by jointly considering the representative MCS performance metrics (i.e. coverage, latency, and accuracy). Focusing on the vehicle-based crowdsourcing, Zhang et al. [25] formulated the worker recruitment problem as a bi-objective optimization model w.r.t. query reliability and sensing coverage.

Different from the above-mentioned studies, we followed the research line of Sparse MCS and focused on providing a comprehensive modeling of the inconstancy sensing cost. The task assignment problem is simplified as a bi-objective cell selection process (i.e. informativeness and sensing cost). Also, we discussed the solvable conditions and assumptions of our model. Moreover, we proposed two heuristic cell selection strategies and evaluated their performance on four city-level real-world datasets.

B. Sparse MCS

As almost all physical conditions monitored are continuous, sensory data generally exhibit strong spatial-temporal correlation, thus the environment ground truth matrix [26] often has a low-rank feature.

With this insight, Wang et al. [27] proposed to use the overall inference error, rather than the sensing area coverage, as the data quality metric. In such MCS tasks, compressive sensing has become the *de facto* choice of the inference algorithm. Then, Wang et al. [13] defined the specific MCS problem as Sparse MCS and discussed the challenges as well as the opportunities from three aspects. Next, Wang et al. [28] extended the Sparse MCS to dynamically select a small set of sub-areas for sensing in each timeslot for multi-task scenarios. Later, Wang et al. [29] also added a privacy protection mechanism into Sparse MCS. In the above-mentioned Sparse MCS, since the entropy-based cell selection only chooses the cell which is the most uncertain at that moment, and ignores whether the current selection would help the inferring in the future or not. Thus, Wang et al. [30] and Liu et al. [15] proposed the deep reinforcement learning-based cell selection method in Sparse MCS. This method proves to narrow the gap with the optimal solution.

Different from the previous works, Xie et al. [17] found that the matrix completion method saves more samples than the vector-based compressive sensing methods. They proposed an active sparse MCS scheme which includes a bipartite-graph-based sensing scheduling scheme to actively determine the sampling location positions in each upcoming time slot, and a bipartite-based matrix completion algorithm to robustly and accurately recover the unsensed data in the presence of sensing and communication errors. Recently, Liu et al. combined the deep reinforcement learning-based cell selection method with practical user recruitment model to deal with the data inference improvement [31]. In this paper, we also focus on cell selection process and aim to incorporate cost-diversity into our cell selection strategies.

C. Techniques in Subset selection

It is a fundamental problem to select the optimal subset from a large set of variables in various learning tasks, such as feature selection, sparse regression, and dictionary learning [32]. Obviously, our research problem of selecting partial sample locations, and inferring missing data in the unsensed cells can also be transformed into a subset selection problem. The subset selection problem is, however, generally NP-hard [33].

To address this problem, previous techniques can mainly be categorized into two branches, greedy algorithms and convex relaxation methods. Generally, greedy algorithms iteratively select or abandon one instance that makes the objective currently optimized [34]. Though the greedy nature of the generalized greedy algorithm can guarantee an efficient fixed runtime, but limits its performance at meantime. Besides, convex relaxation methods usually replace the set size constraint (i.e., l_0-norm) with convex constraints, then find the optimal solutions to the relaxed problem, which however can be distant to the true optimum. In recent years, Qian et al. proposed an evolutionary algorithm, namely POSS, treated the subset selection problem as a bi-objective optimization problem that optimizes some specific criterion and the subset size simultaneously. This algorithm is an anytime method that can use more time to find better solutions.

Note that Huang et al. [35] was the first work leveraging the POSS algorithm to solve the bi-criteria feature acquisition in low-rank active matrix approximation problems. Inspired by their work, we can formulate our cell selection process into a subset selection problem with bi-objective as well (maximize the information in selected cells and minimize the sample costs at meantime). But in our paper, we proposed two heuristic strategies, not only the POS strategy. Also, the cost budget in [33] and [35] is constant, but the cost budget in a selection of our paper is dynamic because we set it as the biggest cost value of unsensed cells plus one. Moreover, we also adjusted the definition on the information goals and its form.

III. System Model and Problem Statement

In this section, we first present the three-stage model to describe the activity in a Sparse MCS platform: cell selection, quality assessment, and data inference, and then introduce the quality assessment method, i.e. *leave-one-out statistical analysis* (*LOO-SA*) and the data inference method, i.e. compressive sensing (CS) in brief. Finally, we discuss the cost-quality beneficial cell selection solution in Sparse MCS using a running example. Table I shows the main concepts and notations used in this paper.

TABLE I
Main Notations and Concepts

Notations	Explanations
n	The sensing campaign is divided into n cycles
T	The theoretical repeat iterations in POMC
$\mathcal{V}$, m	The subarea set (cell), with m subareas (cell)
$G_{m\times n}$, $\hat{G}_{m\times n}$	The ground truth data matrix of m subareas in n sensing cycles; and the inferred one
$S_{m\times n}$	The cell selection matrix, which marks whether a subarea is selected (its entry equals to 0 or 1)
$M_{m\times n}$	The actual measurement data matrix, which records the actual sensed data $M = G \circ S$
$f_1(S)$	The information estimation algorithm, which estimates the information of selected cells
$f_2(S)$	The cost estimation algorithm, which estimates the dynamic sample cost of each cell
$I_{m\times n}$, $C_{m\times n}$	The information matrix and the cost matrix
ε_k	The overall sensing error in each cycle k: $\varepsilon_k = \text{error}(\hat{G}[:,k], G[:,k])$
$error()$ function	The specific metric for calculating the overall sensing error and in this paper it is mean absolute error: $error(\hat{G}[:,k], G[:,k]) = \frac{\sum_{i=1}^{m} \lvert\hat{G}[:,k]-G[:,k]\rvert}{m}$
(ϵ, p)-quality	It is data quality requirement for n cycles and the quality guarantee means that in $p \times 100\%$ of cycles, the inference error is not larger than the predefined error bound ϵ [27] . Formally, we have $\lvert\{k \lvert error(G[:,k], \hat{G}[:,k]) \le \epsilon, 1 \le k \le n\}\rvert \ge n \cdot p$
B_{cost}^{all}	The total cost budget for a MCS task over all cycles
B_{cost}^{one}	The cost budget in a selection
c_v, c_b	The measurement cost and the perception cost

A. System model

A typical MCS sensing scenario begins with a sensing task launched by its organizer to obtain fine-grained urban context results, e.g. humidity over a large-scale target area for a long time, as shown in Fig. 2. To provide high-quality sensing services, the target sensing area is divided into m subareas according to the organizer's requirement. In the meantime, the whole sensing campaign is also split into n equaling sensing cycles. For instance, the organizer needs to update the full humidity sensing map once every hour (sensing cycle), and in each sensing cycle, the data quality requirement is that the mean absolute error for the whole area should be less than 1.5% (humidity). To meet the data quality requirement under the constraint of task budget B_{cost}^{one} in each selection, the organizer needs to carefully select subareas to make a tradeoff between the informativeness (i.e. maximize the information to reduce inference error) and the sensing cost (i.e. reduce total costs) in a subarea. After meeting the quality requirement in a sensing cycle, the humidity values of the remaining cells are deduced based on the sensed humidity values of those selected cells. Through this crowd-powered subset sensing approach, the organizer can obtain sufficient data based on the sensing requirement and costs.

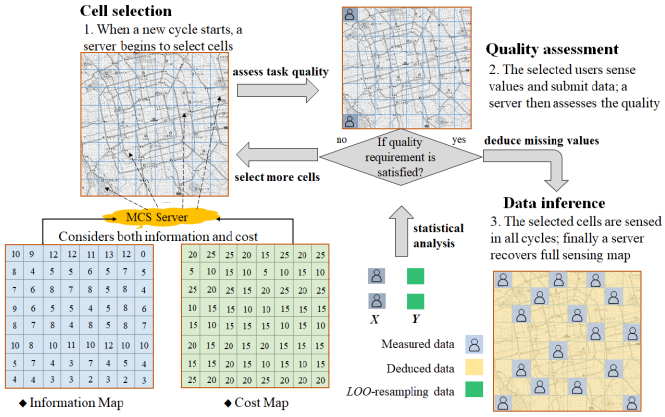


Fig. 2. The general process of the system model

B. Data inference

In Sparse MCS, we often leverage the historical and the current sensed data to infer the data of the remaining unsensed cells. Compressive sensing, as a good choice for inferring the full ground data matrix from the partially collected sensing values with convincing theoretical deviation, has shown its effectiveness in several scenarios [36], [26]. Now we recover the full ground data matrix $\hat{G}_{m\times n}$ based on the low-rank property as follows:

$$\begin{aligned} &\min\ rank(\hat{G}) \\ &s.t.,\ \hat{G} \circ S = M \end{aligned} \tag{1}$$

where $\circ$ denotes the element-wise multiplication and each entry S_{ij} denotes whether the cell i at cycle j is selected for sensing. Thus, S_{ij} equals 0 or 1. Based on the singular value decomposition (SVD) and compressive sensing theory, i.e. $\hat{G} = LR^T$, we convert the above nonconvex optimization problem as follows:

$$\min\ \lambda(||L||_F^2 + ||R||_F^2) + ||LR^T \circ S - M||_F^2 \quad (2)$$

This optimization changes the rank minimization problem (minimize the rank of $\hat{G}$) to minimizing the sum of *L* and *R*'s Frobenius norms. And λ allows a tunable tradeoff between rank minimization and accuracy fitness. To get the optimal $\hat{G}$, alternating least squares [37] procedure is leveraged to estimate *L* and *R* iteratively.

Moreover, strong spatial-temporal correlations can be discovered in the sensing data [38], [39]. Thus, adding the explicit spatiotemporal correlations into compressive sensing, the optimization function can be further formulated as:

$$\begin{aligned}\min\ & \lambda_r(||L||_F^2 + ||R||_F^2) + ||LR^T \circ S - M||_F^2 \\ & +\lambda_s||\mathbb{S}(LR^T)||_F^2 + \lambda_t||(LR^T)\mathbb{T}^T||_F^2\end{aligned} \quad (3)$$

where $\mathbb{S}$ and $\mathbb{T}$ are spatial and temporal constraint matrices respectively; λ_r, λ_s and λ_t control the tradeoff between different correlations. Concerning the spatial and temporal constraint matrix, interested readers are referred to this study [26].

C. Quality assessment

In this paper, *LOO-SA* is used to assess the inference quality. First, a leave-one-out resampling mechanism is implemented to obtain the set of (*inferred*, *true*) data pairs. Then, by comparing the *inferred* data with the corresponding *true* collected data, Bayesian inference or Bootstrap analysis is leveraged to assess whether the current data quality can satisfy the predefined (ϵ, p)-quality requirement.

Leave-one-out is a popular resampling method to measure the performance of many prediction and classification algorithms. Suppose that we collect sensing data from m' out of all the *m* cells, the idea of *LOO* is that for each time, we leave one observation out and infer it based on the rest $m' - 1$ observations by using compressive sensing or interpolation algorithms. After running this process for all m' observations, we get m' predictions accompanying with the m' true observations, as shown in Eq.(4).

$$\mathbf{x} = \langle x_1, x_2, ..., x_{m'} \rangle\ ,\quad \mathbf{y} = \langle y_1, y_2, ..., y_{m'} \rangle \quad (4)$$

Based on the m' (*inferred*, *true*) data pairs, we can use Bayesian inference or Bootstrap analysis to estimate the probability distribution of the inference error ε, e.g. mean absolute error (MAE) to help quality assessment. Actually, satisfying the (ϵ, p)-quality can be converted to calculate the probability of $\varepsilon_k \leq \epsilon$, i.e. $P(\varepsilon_k \leq \epsilon)$, for the current cycle *k*. If $P(\varepsilon_k \leq \epsilon) \geq p$ can hold for every cycle *k*, then (ϵ, p)-quality is expected to be satisfied as a whole. In this paper, two statistical analysis methods, i.e., Bayesian inference [40] and Bootstrap analysis [41], are leveraged for estimating $P(\varepsilon_k \leq \epsilon)$. Different from Bayesian inference (require the error metric is normal distribution), the advantage of Bootstrap is that no assumption on the distribution of the observations needs to be made. Detailed information about Bayesian or Bootstrap analysis can be found in reference [13].

D. Assumptions, problem formulation and use case study

1) Assumptions. We follow the basic assumptions in [16], [27] in our study except for the assumption of sample cost. The unsound assumption in previous Sparse MCS is that each sample has the same cost: its goal is to simply reduce the number of samples while achieving a good recovery accuracy. Since the cost of obtaining a sample depends highly on the location, time, sensing device types, condition of the device, and many other factors of the sample, we break the assumption in previous studies of Sparse MCS and further assume that the cost of obtaining a specific sample of each subarea in different cycles is diverse. To make the cell selection problem more practical, the assumption of cost-diversity allows us to make a tradeoff selection considering both the sample cost and informativeness in a spatial-temporal cell.

2) Problem formulation. Based on the previous system model, assumptions and the brief introduction on compressive sensing and *LOO-SA*, we define our research problem and focus on the cell selection. The cell selection problem can be formulated as Eq.(5): given a MCS task with *m* cells and *n* cycles, the sensing budget B_{cost}^{all}, a sensing matrix inference algorithm $\mathcal{R}$, an information estimation algorithm $f_1(S)$, and a sample cost estimation algorithm $f_2(S)$, the MCS organizer attempts to select a subset of most informative sensing cells under the task budgets during the whole process (use the minimal costs to find the subset cells maximal in information), while satisfying the (ϵ, p)-quality.

$$\begin{aligned}&\arg\min_{\boldsymbol{s}\in\{0,1\}^{mn}} \left(-f_1(\boldsymbol{s}), f_2(\boldsymbol{s})\right), \\ &\text{s.t., } |\{k|\varepsilon_k \leq \epsilon, 1 \leq k \leq n\}| \geq n \cdot p \\ &\text{where } \varepsilon_k = \text{error}(\hat{G}[:,k], G[:,k]), \\ &f_1(\boldsymbol{s}) = \sum_i \sum_j S_{ij} \cdot I_{ij}, \\ &f_2(\boldsymbol{s}) = \sum_i \sum_j S_{ij} \cdot C_{ij} \leq B_{\text{cost}}^{all}\ .\end{aligned} \quad (5)$$

This optimization problem aims to maximize the information estimation function $f_1(\boldsymbol{s})$ and minimizes the cost estimation function $f_2(\boldsymbol{s})$ simultaneously. Here we maximize the information of selected cells by minimizing its negative $f_1(\boldsymbol{s})$. The overall sensing error ε_k and the error metric (mean absolute error) are defined in Table I. Note that we use a Boolean vector $\boldsymbol{s} \in \{0,1\}^{mn}$ to replace the cell section matrix $S_{m\times n}$, where the $i + m \times j$ bit $s_{i+m\times j} = 1$ means that the entry S_{ij} equals 1. In this paper, we will not distinguish $\boldsymbol{s} \in \{0,1\}^{mn}$ and its corresponding representation $S_{m\times n}$. As we cannot foresee the ground truth matrix $G_{m\times n}$ for a MCS task, it is impossible to obtain the optimal cell selection matrix $S_{m\times n}$ in reality. To overcome the difficulties, we propose a novel cell selection method, which leverages an iterative process to select sensing cells in each cycle, with details elaborated in the following section.

3) Use case study. Fig. 3 shows the basic idea of our proposed cell selection process in a sensing cycle. Suppose the target area contains six cells and the sixth sensing cycle start currently; at first, no sensing data is collected in the sixth cycle

(Fig. 3(1)). Our proposed cost-quality cell selection method (e.g. take the POS strategy as an example) works as follows:

(1) First, under the given cost budget (e.g. 2 CNY) of a selection, our strategy lists all possible solution combinations (i.e. here we have 7 possible solutions within the cost budget and they are single cell 1 to 6 and the combination of cell 1 and 6), compares the solutions by considering the tradeoff between information and sample costs, selects the first two salient cells (cell 1 and cell 6) under the budget, i.e. 2 and allocates the sensing tasks to two mobile users in cell 1 and cell 6 respectively. Mobile users perform the sensing tasks and return the sensing data to the MCS server (Fig. 3(2)).

(2) Then, given the collected sensing data, the quality assessment module decides if the data quality satisfies the predefined (ϵ, p)-quality requirement. If the data quality does not meet the quality requirement, we have to select more cells for sensing (choose cell 5 to sense in Fig. 3(3)). In this way, our strategy continues to allocate tasks to new cells and collects sensing data (illustrated in step (1), and more details will be introduced in the section of cell selection, until the quality of the collected sensing data satisfies the predefined requirement.

(3) Given the collected sensing values, the compressive sensing module module infers the values of unsensed cells.

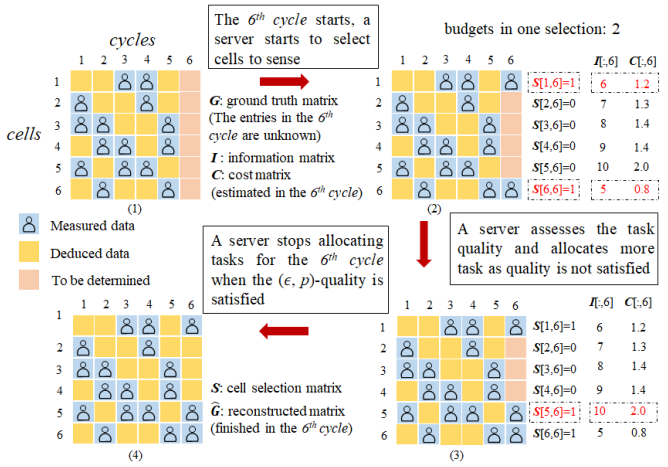


Fig. 3. An example of our cell selection process (6 cells and 6 cycles)

IV. Cost-Quality Beneficial Cell Selection in Sparse MCS

In this section, we will introduce the information modeling, sample cost estimation and cost-quality beneficial cell selection in sequence.

A. Information modeling

In traditional active learning, if the model is less certain about the prediction on an instance, then the instance is considered to be more informative for improving the model and will be more likely to be selected for label querying [35]. Inspired by this idea, we can leverage a reward criterion or a value criterion [42] to estimate the informativeness of an entry, i.e. a spatiotemporal cell in the matrix G. The challenge here is how to quantify the informativeness of an instance in a subarea for recovering the entries in other subareas. Obviously, the entropy-based (e.g. QBC) or mutual-information-based method (e.g. Gaussian process-based mutual information) belongs to a reward criterion, which indicates what is good in an immediate sense; while the value-function-based method (reinforcement learning based method) considers what is good in the long run. Though mutual-information-based method and value-function-based method can better quantify which cell is more informative, they require sufficient historical data to compute the informativeness of a cell. Also, they are unable to be applied in a fresh task when no sufficient data is acquired. Therefore, in this paper, we only take the simple but general method, i.e. QBC as the information estimation method in the evaluation section.

QBC originates from such an idea that if the variance of an entry is large, it implies that the entry cannot be certainly decided by the algorithm, and thus may contain more useful information to recover the estimated full sensing matrix. QBC framework quantifies the prediction uncertainty based on the level of disagreement among an ensemble of matrix completion algorithms. Specifically, a committee of matrix completion algorithms is applied to the partially observed data matrix to impute the missing values. The variance of prediction (among the committee members) of each missing entry is taken as a measure of uncertainty of that entry. In this paper, the committee consists of several commonly used inference algorithms, including CS, STCS, K-Nearest-Neighbors (KNN), and SVD.

It is assumed that the committee includes a set of L inference algorithms. In a sensing cycle j, the already selected cells with measurements in this cycle are denoted by S_j ($S_j \in \mathcal{V}$). The sensor measurements in these selected locations are represented by $\chi_{S_j} = x_{S_j}$. By using one of the inference algorithms, we have $\hat{G}(:, j) = \mathcal{R}_l(x_{S_j})$. For an unsensed cell $\upsilon \notin S_j$, the informativeness of this cell can be formulated as:

$$I_{\upsilon,j} = \sum_{l=1}^{L} \left(\hat{G}_l(\upsilon, j) - \bar{G}(\upsilon, j)\right)^2 / L \tag{6}$$

where $I_{\upsilon,j}$ represents information of unsensed cell υ in time cycle j; $\bar{G}(\upsilon, j)$ denotes the average value predicted by the committee; $\hat{G}_l(\upsilon, j)$ is the predicted value of the l-th matrix completion algorithm in cell υ.

B. Cost estimation

Since the cost of obtaining a specific sample in practical MCS systems depends highly on the location, time, condition of the device, human expectation, density of participants, moving distance, and many other factors, we will consider cost-diversity in the cell selection process. Different from the cost modeling in [18], in the following, we first discuss a more broad types of costs occurring in MCS systems, including routing cost, measurement cost, perception cost and their combination. Then, we introduce a new cost function to estimate the sensing cost. Finally, we present several challenges of incorporating costs into Sparse MCS, thus we need to make some compromises on the implementation of estimation on costs.

1) Cost factors: In practice, different types of costs often occur in MCS systems, including but not limited to (1) routing cost; (2) measurement cost; (3) perception cost.

Routing cost. Consider such a scenario, when mobile users in some cells are insufficient, so both the costs of moving from the present location to the target location and that of making measurements at subareas need to be considered. We use the cost function $c(\mathcal{V})$ to denote such a cost, defined as $c(\mathcal{V}) = c_R(\mathcal{V}) + \sum_{v\in\mathcal{V}} c_v$, where $c_R(\mathcal{V})$ is the cost of the shortest walk to visit each selected subareas in $\mathcal{V}$ at least once. Note that $c_R(\mathcal{V})$ is generally non-submodular and cannot be exactly computed in polynomial time. It will be discussed in detail in the following subsection.

Measurement cost. We use c_v to denote the cost of collecting measurements in each subarea $v \in \mathcal{V}$. Generally, this kind of cost consists of *energy consumption* and *data consumption* on sensing devices as well as data management cost. Devices consume energy in both measuring and reporting a sample, e.g. locate a GPS signal and report position. This cost depends on the location as well as the status of the device. The reporting cost may depend on the network, i.e. WiFi, 3G or 4G, the signal strength, variability to the network, and the congestion level. Meanwhile, the reporting may incur cellular data cost when using cellular networks. Also, the submitted data stored in Cloud or network and the quality control of data incur a management cost.

Perception cost. Finally, mobile users may have different perceptions of a given cost. In other words, this cost is a subjective evaluation of the provided services. For example, a user carrying on a smartphone with a full battery may not care about the energy consumption for GPS locating to be a large cost, whereas other users may be more sensitive to the same amount of energy usage when they are running out of the battery of smartphones. Such perception-based cost adjustments c_b should be considered as they are important to user experience in MCS applications.

2) Cost functions: Here we introduce a cost function to estimate the value of sample costs when the actual cost is hard to acquire. Since it is assumed that sufficient participants exist in each subarea waiting for recruitment in previous Sparse MCS, the routing cost across different cells will thus not be included in this paper. To estimate the measurement cost, we can conduct outdoor measurements of GPS energy consumption using smartphones at hundreds of subareas across the target region and record the data consumption of 3G/4G/5G cellular network at those locations in the meantime.

Further, we naturally consider the remaining battery level of a device as a type of "perception cost": the lower the remaining battery, the more valuable it is, the higher cost it should be assigned. We define $c_b = B^{1-b}$ as the perception-based cost for the remaining battery, where *b* is the ratio of the remaining battery, and *B* is a constant. In particular, as *b* goes to zero, the cost is high and approaches *B* quickly. The intuition is that when *b* is large, mobile users are not sensitive and thus the measurement cost dominates. On the other hand, when *b* is small, users are sensitive and thus this factor will contribute a lot to the final cost. Therefore, we choose a multiplication combination of measurement cost (initial cost function) and perception cost (multiplier, forming a dynamic cost function) as the overall sensing cost function $c_v \cdot c_b$.

Prior study [18] proved that synthetic cost maps based on the overall cost estimation function are also feasible for performance evaluation when it is hard to conduct practical measurement and estimation. Therefore, we generate three synthetic cost maps based on our proposed cost function, and they are i.i.d with dynamic cost map, monotonic with dynamic cost map, and spatial correlated with dynamic cost map. Here, we take the final dynamic cost map in traffic dataset as an example, the cost distribution on different subareas over time (in a day) is exhibited in Fig. 4. In the rest of the paper, we use CT1, CT2 and CT3 to refer to i.i.d with dynamic cost map, monotonic with dynamic cost map and spatial correlated cost map, respectively. As we see in Fig. 4, the three sampling cost maps present different changing characteristics (randomness in CT1, monotonicity in CT2 and spatial correlation in CT3), with darker color indicating larger cost.

3) Challenges of estimating costs: (1) Difficult to estimate cost accurately. Since real cost are hard to obtain, a cost estimation function is often required. Due to the cost-diversity, a simple cost estimation function is hard to estimate the value of sample cost accurately. Though in present practices, multi-factor regression models are trained to estimate the current cost of the operation and the influence of prior cost observations is also considered, this design is still far from practical values. How to design an estimator to give an entirely accurate estimation of the sensing cost is not the focus of this paper. Thus, we leverage the synthetic cost maps generated by the cost function $c_v \cdot c_b$. (2) Difficult to accomplish estimation on certain types of cost in polynomial time, e.g. routing cost. Unlike other typical additive cost constraints, such routing planning costs are themselves NP-hard to evaluate [43]. Therefore, to ensure the algorithm efficiency in cell selection, necessitating approximation in practices.

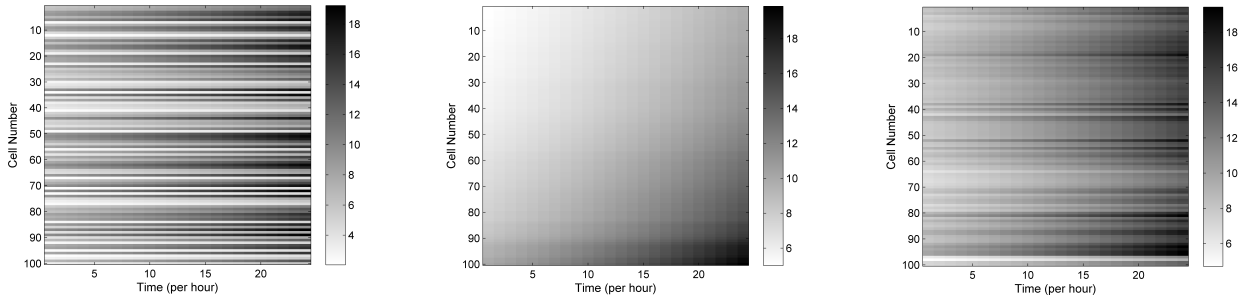
C. Discussions on objective functions

We first give the definition of submodularity and then discuss whether the objective functions have this property when certain conditions are satisfied. If Ω is a finite set, a submodular function is a set of function $f : 2^{\Omega} \rightarrow \mathbb{R}$, where 2^{Ω} denotes the power set of Ω, which satisfies the following conditions:

For every $X, Y \subseteq \Omega$ with $X \subseteq Y$ and every $x \subseteq \Omega \backslash Y$ we have that $f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y)$.

(1) Discussion on f_1

Since $f_1(\boldsymbol{s})$ is the informativeness of the selected cell set, it is nonnegative and non-decreasing when a new element is added. Meanwhile, we notice that $f_1 : 2^{\boldsymbol{s}} \rightarrow \mathbb{R}$ is a submodular function for satisfying the definition. If the informativeness of a cell is estimated by the QBC method, $f_1(\boldsymbol{s})$ would be the entropy of the set of random variables $\boldsymbol{s}$. Then, $f_1(\boldsymbol{s})$ would be a monotone submodular function. However, if the informativeness of a cell is estimated by the mutual-information method or the value-function based method, then $f_1(\boldsymbol{s})$ would be a non-monotone submodular function.



(a) i.i.d with dynamic cost map (Cost Type 1, CT1) (b) Monotonic with dynamic cost map (Cost Type 2, CT2) (c) Spatial correlated with dynamic cost map (Cost Type 3, CT3)

Fig. 4. Three different types of cost maps on the traffic dataset

(2) Discussion on f_2

If routing cost is not considered, no matter sensing cost is constant or inconstant, $f_2(\boldsymbol{s}) = \sum C_{ij}$ is called a linear function. Additionally, since C_{ij} is nonnegative, then $f_2(\boldsymbol{s})$ is a monotone submodular function. If routing cost is considered and denoted by the number of edges (edges formed by the vertices of subareas), then $f_2(\boldsymbol{s})$ is a non-monotone submodular function. If routing cost is denoted as the cost of shortest walk to visit each selected subareas in this paper, then $f_2(\boldsymbol{s})$ is non-submodular and non-monotone.

The cell selection problem in this paper is definitely NP-hard (the proof can be found in the reference [32] since it is a subset selection problem) and sometimes hard to solve when estimation on routing cost is considered. Since we have assumed sufficient participants in each subarea (f_2 is monotone submodular) and f_1 is estimated by QBC (f_1 is monotone submodular), the cell selection problem is solvable by leveraging the following Algorithm 1 and Algorithm 2.

D. Cost-quality beneficial cell selection

1) The cost-quality beneficial cell selection strategy: With the cost and informativeness estimation method above, in this section, the diversity of sample cost is incorporated into the cell selection process, and we propose two selection strategies, namely GCB-GREEDY and POS, to balance the two objectives at meantime: minimize the sensing cost and maximize the informativeness in the collected cells. The detailed strategies of cell selection are formulated as follows.

(1) The generalized cost-benefit greedy selection strategy (GCB-GREEDY)

The cell selection process in previous Sparse MCS can be described as a typical subset selection problem. Generally, the subset selection problem tries to select a subset S_j (salient cells) from the subarea set $\mathcal{V}$ with an objective function $f_1(S)$ (information function) and a constraint of the subset size (select one by one) in each cycle *j*. Therefore, the previous cell selection problem can be formalized as:

$$\arg\max_{S_j \subseteq \mathcal{V}} \ f_1(S_j) \qquad s.t. \quad |S_j| \leq B_{size} \tag{7}$$

where $|\cdot|$ denotes the size of a set; B_{size} is the maximum number of selected elements (the stopping criterion is decided by *LOO-SA*). But in a cost-quality beneficial selection, the constraint of subset size should be transformed into the budget constraint as $f_2(S_j) \leq B_{cost}^{one}$. At the core of the GCB greedy algorithm is the following heuristic: in each iteration *k*, add to the set S_j an element υ_k such that:

$$\upsilon_k \leftarrow \arg\max_{\upsilon \in \mathcal{V} \backslash S_j^{k-1}} \frac{f_1(S_j^{k-1} \cup \upsilon)}{f_2(S_j^{k-1} \cup \upsilon)} \tag{8}$$

where S_j^0=$\varnothing$ and S_j^k=$\{\upsilon_1,...,\upsilon_k\}$. The number of cells in one selection depends on the information and cost budget. Since the routing cost is ignored due to the sufficient participants assumption in the cost function, our cell selection problem is transformed into a problem of maximizing a monotone submodular function f_1 with a monotone approximate cost constraint f_2. The corresponding GCB greedy algorithm is shown in Algorithm 1. It iteratively selects one subarea υ to sense such that the ratio of the marginal gain on f_1 and f_2 by adding υ is maximized.

Algorithm 1 The GCB-greedy-based Cell Selection Algorithm.

Input:
A monotone objective function, f_1;
A monotone approximate cost function, f_2;
The budget constraint, B_{cost}^{one}.
Output:
The solution $S_j \subseteq \mathcal{V}$ with $f_2(S_j) \leq B_{cost}^{one}$.
Let $S_j = \varnothing$ and $\mathcal{V} = \mathcal{V}'$;
repeat
  $\upsilon^* \leftarrow \arg\max_{\upsilon \in \mathcal{V}'} \frac{f_1(S_j \cup \upsilon) - f_1(S_j)}{f_2(S_j \cup \upsilon) - f_2(S_j)}$;
  if $f_2(S_j \cup \upsilon^*) \leq B_{cost}^{one}$ **then** $S_j = S_j \cup \upsilon^*$ **end if**;
  $\mathcal{V}' = \mathcal{V}' \backslash \{\upsilon^*\}$;
until $\mathcal{V}' = \varnothing$
Let $\upsilon^* \leftarrow \arg\max_{\upsilon \in \mathcal{V}; f_2(\upsilon) \leq B_{cost}^{one}} f_1(\upsilon)$;
return $\arg\max_{S'_j \in \{S_j,\ \upsilon^*\}} f_1(S'_j)$ and S'_j.

(2) The Pareto optimization selection strategy (POS)

Inspired by the solutions in [32], the subset selection problem in (7) can be reformulated as optimizing a binary vector. We introduce a binary vector $\boldsymbol{s} \in \{0,1\}^m$ to indicate the subset membership, where $s_i = 1$ if the *i*-th element in $\mathcal{V}$ is selected in a sensing cycle, and $s_i = 0$ otherwise. So

the cell selection problem can be formulated as a bi-objective minimization model:

$$\begin{aligned}&\arg\min_{\boldsymbol{s}\in\{0,1\}^m}\left(-f_1(\boldsymbol{s}),f_2(\boldsymbol{s})\right),\\&f_1(\boldsymbol{s})=\begin{cases}-\infty & \text{if } \boldsymbol{s}=\{0\}^m \text{ or } f_2(\boldsymbol{s})\geq B_{\text{cost}}^{one}+1\\ \sum_i\sum_j S_{ij}\cdot I_{ij} & \text{otherwise.}\end{cases}\\&f_2(\boldsymbol{s})=\sum_i\sum_j S_{ij}\cdot C_{ij}\end{aligned}\tag{9}$$

where $|\boldsymbol{s}|$ denotes the number of 1s in s; S_{ij} denotes the entry in cell selection matrix $S_{m\times n}$; I_{ij} represents the information of cell *i* in sensing cycle *j*; C_{ij} is the approximate sample cost of cell *i* in sensing cycle *j*; B_{cost}^{one} is the cost budget in one selection, which is set as the maximal cost value of unsensed cells (this kind of dynamic cost budget has never been considered before); f_1 is set to $-\infty$ to avoid trivial or over-cost solutions. We use the value $B_{\text{cost}}^{one}+1$ instead of B_{cost}^{one} in the definition of f_1 as this gives the algorithm some look ahead for larger constraint bounds. However, every value of at least B_{cost}^{one} would work for our theoretical analysis. The only drawback would be a potentially larger population size which influences the runtime bounds. The bi-objective optimization model performs active selection to maximize the informativeness and meanwhile to minimize the sample costs of the selected cells. We then employ a recently proposed Pareto Optimization for Monotonic Constraints (POMC) algorithm [33] to solve this problem. POMC is an evolutionary style algorithm, which maintains a solution archive, and iteratively updates the archive by replacing some solutions with better ones. It is also known as Global SEMO in the evolutionary computation literature [44], shown in Algorithm 2.

Algorithm 2 The POMC-based Cell Selection Algorithm.

Input:
A monotone objective function, f_1;
A monotone approximate cost function, f_2;
The budget constraint, B_{cost}^{one};
The number of iterations, *T*.
Output:
The solution $\boldsymbol{s}\in\{0,1\}^m$ with $f_2(\boldsymbol{s})\leq B_{cost}^{one}$.
Let $\boldsymbol{s}=\{0\}^m$ and $P=\{\boldsymbol{s}\}$;
Let *t* = 0;
while $t<T$ **do**
  Select $\boldsymbol{s}$ from *P* uniformly at random;
  Generate $\boldsymbol{s}'$ by flipping each bit of $\boldsymbol{s}$ with probability $1/m$;
  if $\nexists z\in P$ such that $z\succ\boldsymbol{s}'$ **then**
    $P=(P\backslash\{z\in P|\boldsymbol{s}'\succcurlyeq z\})\cup\{\boldsymbol{s}'\}$;
  end if
  $t=t+1$;
end while
return $\arg\max_{\boldsymbol{s}\in P;f_2(\boldsymbol{s})\leq B_{cost}^{one}} f_1(\boldsymbol{s})$ and $\boldsymbol{s}$.

2) The cell selection algorithm for a MCS task: The above two strategies are proposed to compute the approximate optimal solution for only one selection. Considering *n* cycles and *m* cells in our problem, we summarize the pseudo code of the proposed algorithm in Algorithm 3. When a new sensing cycle starts, the MCS server needs to update the cost map at first. Then, the information of unsensed cells needs to be computed by QBC. Next, we set the cost budget in one selection at the maximal sample cost (or a little bit larger). Consequently, we adopt different cell selection strategies to solve the subset selection problem with cost constraints. After that, the MCS server recruits participants to collect actual sensing data in the selected cells and aggregates the collected data to judge if more cells are required to sense. If the predefined quality requirement is not satisfied, we repeat the steps 6-9 until the predefined quality requirement is satisfied. The quality requirement satisfied indicates that the MCS server can stop sensing in this cycle and move to the next cycle. The MCS server repeats the above steps until (ϵ,p)-quality in all time cycles is satisfied. Finally, we can deduce the unsensed data through compressive sensing based on sensed data.

Algorithm 3 The cost-quality beneficial cell selection algorithm for a MCS task.

Input:
The budget constraint, B_{cost}^{all};
Predefined quality requirement, (ϵ,p)-quality;
The sensing matrix reconstruction algorithm, $\mathcal{R}$;
The cost map, $C_{m\times n}$; The error metric, e.g. MAE or CE.
Output:
The Inferred full ground data matrix, $\hat{G}_{m\times n}$.
repeat
  new sensing cycle *t* starts, update the cost map for the this cycle;
  repeat
    compute the informativeness of the un-measured sub-areas through Eq.(6);
    determine the cost budget for a batch of chosen cells in a selection;
    solve the subset selection problem considering both information and sample costs through different cell selection strategies;
    send participants to collect sensing data in the selected cells;
    assess task quality in this time cycle;
  until the predefined quality requirement is satisfied;
until the predefined quality requirement in all time cycles is satisfied;
return The estimated full ground truth matrix $\hat{G}_{m\times n}$.

3) Computation complexity: Since cell selection strategies depend on informativeness modeling (QBC is selected in this paper), and thus QBC contributes much to the running time. Due to the fact that the runtime of QBC is mainly spent on using different inference algorithms to reconstruct the sensing matrix, then the complexity of QBC can be formulated as $O(\sum_l T_{\mathcal{R}_l})$ if the computation complexity of a reconstruction algorithm $\mathcal{R}_l$ is $T_{\mathcal{R}_l}$. Besides, due to the characteristics of different strategies, their runtime performances are widely divergent. The greedy nature of the GCB-greedy algorithm results in itself an efficient fixed time algorithm. While POMC is an anytime randomized iterative algorithm, it needs to spend more time than the greedy algorithm to find the best feasible

solutions. More specifically, the runtime of POMC depends on the setting of parameter T.

V. Analysis of cost-quality aware Sparse MCS assisted urban sensing and actuation

For urban computing [45], [46], traditional practices usually depend on specialized infrastructure, e.g. surveillance cameras, air quality stations, which incurs a high cost for deployment and maintenance. With the advent and development of seamless connections among machines, smart things and humans, it is an emerging trend that a governor or a service initiator leverages the power of crowds, e.g. mobile users and smart things, to monitor what is happening in a city, understand how the city is evolving, and further take actions to enable a better quality of life [47]. In this paper, we offer a governor with the proposed cost-quality beneficial Sparse MCS approach to sense the urban context in a more cost-beneficial way with high-quality sensed data and inferred data.

(1) Benefits to urban context sensing

Nowadays, real-time information in a city, for instance, the shortage of parking bothers the managers and causes severe societal problems, like traffic congestion and environmental pollution. In previous practices, a governor would employ dedicated staff and leverage expensive resources to monitor and report the parking occupancy situation, which incurs large operational costs. Meanwhile, note that under a large-scale target area, we usually have many subareas for a fine-grained result and need to recruit a large number of participants, which also costs a lot. Alternatively, we can leverage the cost-quality aware cell selection approach proposed in this paper to recruit only a few number of mobile users to collect real-time parking availability information in some subareas and report the collected information to the centralized server. Then, the server would exploit the compressive sensing or matrix completion techniques to recover the information in unsensed subareas. Other examples, such as passenger flows in a target area, traffic situation, and air quality, are also important issues to a governor as well as the citizens and can be sensed by our proposed crowd-powered way. Therefore, our proposed crowd-powered urban context sensing can fulfill the task of sensing in large urban regions with less cost and higher efficiency.

(2) Benefits to urban context actuation

The further intention of a governor or a service initiator is to adopt measures or impose influence on the urban context by leveraging the collected and inferred information. The management mode of a city would be changed to optimize different smart systems (e.g. smart parking, intelligent transit) and enable better quality of life (e.g. recommendation of parking spot, reschedule travel plans). For instance, due to the outbreak of COVID-19, citizens are required to maintain social distance for a certain period of time. But if the collected information in a subarea about passenger flow index exceeds the predefined threshold, the local governor will suggest citizens in other regions to not travel to this region and take strict isolation measures in this region to reduce the flow index. Other examples, such as engaging users to re-balance shared bikes, encouraging citizens and private cars to assist package delivery, suggesting vehicles to take other routes when meeting traffic congestion are also typical actuation applications by leveraging the collected information. In this crowd-powered paradigm, the efficiency of the current smart city systems will be largely improved. It reveals the importance of the information supported by our proposed crowd-powered sensing paradigm.

VI. Performance evaluation

In this section, we evaluate our proposed strategies on four sensing projects, which contain various types of sensed data, including parking occupancy rate, passenger flow index, traffic speed and humidity data.

A. Datasets and the inherent features

In this paper, we adopt four real-life sensing datasets, Birmingham-Parking [48], DataFountain competitions[1], TaxiSpeed [49] and SensorScope [50] to evaluate the applicability of our following proposed algorithms. The datasets contain various types of sensed data in representative IoT applications, like parking occupancy rate, flow index, traffic speeds, and humidity. Though some of the data in these datasets are collected by sensor networks or static stations, mobile users can also sense the data by using smartphones, as shown in studies [3], [51]. The detailed statistics of the four datasets are shown in Table II and their distributions are shown in Fig. 5.

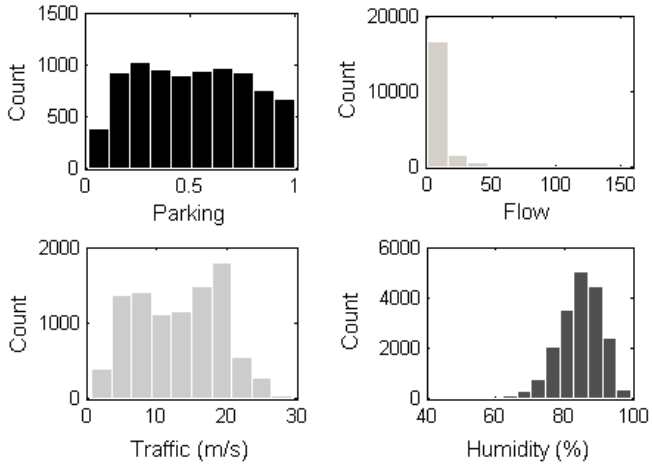


Fig. 5. Data distribution of different urban sensing datasets

Parking (occupancy rate): The car park occupancy rate readings are sensed in the Birmingham-Parking project, collected from 32 different car parks for two months and 16 days with a sensing cycle at 60 minutes. Since the occupancy rates are only recorded for eight hours every day, a target area that has 30 car parks with valid readings (from 8:00 am to 4:00 pm)

[1]DataFountain competitions: DataFountain is the designated platform for the 2016 CCF Big Data and Computing Intelligence Competition, which aims to build China's most influential and authoritative data science and big data analysis and processing platform. The Flow (passenger index) dataset provided by DataFountain is pre-processed through a mathematical model.

TABLE II
STATISTICS OF FOUR EVALUATION DATASETS

	Parking	**Flow**	**Traffic**	**Humidity**
Project	Birmingham-Parking	DataFountain competitions	TaxiSpeed	SensorScope
City	Birmingham	Beijing	Beijing	Lausanne
Cycle length	1 hour	1 hour	1 hour	0.5 hour
Duration	76 days	12 days	4 days	7 days
Cell size	car parks	200m × 200m	road segments	50m × 30m
Number of cells	30	68	100	57
Mean ± Std.	0.5155 ± 0.2597	8.84 ± 13.68	13.01 ± 6.97 m/s	84.52 ± 6.32 %

are leveraged. In this scenario, we take the car parks as the cells.

Flow (passenger index in a region): The dataset is provided by the DataFountain competitions for predicting the future passenger index in Beijing. Specifically, the flow index readings are sensed during the outbreak of the COVID-19 from 100 different types of key regions. The sensing lasts for 30 days from 2020-01-17 to 2020-02-15 with a collecting cycle at 60 minutes and the target region is initially divided into 997 cells with an equal size at $200m \times 200m$. In this paper, only 68 cells with valid values during 12 days are leveraged.

Traffic (speed): The speed readings of taxis are collected for road segments in the TaxiSpeed project in Beijing. The project lasted for 4 days from 2013-09-12 to 2013-09-15. Specifically, this dataset contains more than 33,000 trajectories collected by GPS on taxis. And each sensing cycle lasts for 60 minutes. According to [49], we consider the road segments as the cells, and a target area that has 100 road segments with valid sensed values is selected.

Humidity: The humidity readings are sensed in the SensorScope project, collected from the EPFL campus with an area about $500m \times 300m$ for 7 days (from 2007-07-01 to 2007-07-07). Each sensing cycle lasts for 30 minutes. For our experiments, we divide the target area into 100 cells with each cell size $50m \times 30m$. Since only 57 cells are deployed with valid sensors, we just utilize the sensed data at the cells with valid readings.

In these datasets, the mean absolute error is chosen as the metric to evaluate the inference quality. Also, the datasets used in this paper come from publicly available data on the Internet. After the careful check by the authors, there are no user privacy issues.

Inherent features in the urban sensing data are the prerequisite for spatial-temporal compressive sensing. To ensure the validity of our proposed models and algorithms, we need to conduct a set of experiments on these datasets to discover the strong spatial-temporal correlations. It is the basis and premise of this research. ***Results show that the urban sensing data matrix could have a low-rank approximation, certain temporal stability, and high spatial correlation.***

(1) Low-rank feature

Generally, there often exists an inherent correlated structure or redundancy in long-time urban sensing data. Thus, we apply singular value decomposition (SVD) to examine whether the ground truth sensing matrix has a good low-rank structure. Any ground truth data matrix $G_{m\times n}$ can be decomposed as:

$$G = U \sum V^{tr} \tag{10}$$

where V^{tr} is the transpose of V (a $n \times n$ unitary matrix), U is a $m \times m$ unitary matrix, and $\sum$ is a $m \times n$ diagonal matrix with the diagonal elements σ_i (i.e. singular values) organized in the decreasing order. The rank of a matrix, denoted by *r*, is equal to the number of its non-zero singular values. Specifically, a low-rank matrix means that its matrix rank $r \ll \min\{m, n\}$.

According to principal components analysis, a low-rank matrix has the character that its top *k* singular values occupy the total or near-total variance $\sum_{i=1}^{k} \sigma_i^2 \approx \sum_{i=1}^{r} \sigma_i^2$. Thus, we use the fraction of the total variance captured by the top *k* singular values as the evaluation metric:

$$g(k) = \sum\nolimits_{i=1}^{k} \sigma_i^2 / \sum\nolimits_{i=1}^{r} \sigma_i^2 \tag{11}$$

Fig. 6(a) plots the fraction of the total variance captured by the top *k* singular values as the *k* varies for different urban sensing data. We can find that the top 1.75%-13.3% singular values include over 98% variance in the real datasets. The result indicates that the urban sensing data has a good low-rank approximation.

(2) Temporal stability feature

Temporal stability indicates how the measured data changes over time. In urban sensing, some measured data, e.g. humidity and temperature, usually change slowly over consecutive time slots. But other urban context data may not have this feature. Thus, to reveal the natural phenomenon and check if this feature exists in different urban sensing data, we analyze the datasets in time dimension between each pair of adjacent time measurements at a location.

The temporal stability feature at subarea *i* and time slot *j* is computed by the normalized difference values between adjacent time slots $\Delta tsf(i, j)$:

$$\Delta tsf(i,j) = \frac{|G(i,j) - G(i,j-1)|}{\max\limits_{1\le i\le m, 2\le j\le n} |G(i,j) - G(i,j-1)|} \tag{12}$$

where *i* varies from 1 to *m*, *j* varies from 2 to *n* (*n* is the number of time slots of interest), and $\max_{1\le i\le m, 2\le j\le n} |G(i,j) - G(i,j-1)|$ is the maximal difference of the urban sensing data captured in any two consecutive time slots.

The Cumulative Distribution Function (CDF) of $\Delta tsf(i, j)$ is plotted in Figure 6(b). The X-axis represents the normalized difference between values in two consecutive time slots, i.e. $\Delta tsf(i, j)$. The Y-axis denotes the cumulative probability. It

is observed that more than 60% of $\Delta tsf(i,j)$ values are very small (<0.1) for the four different datasets. Even in the worst case, the traffic values between two consecutive time slots mostly (>90%) only change a little (<0.3). These findings indicate that the real urban sensing data is temporally stable.

(3) Spatial correlation feature

Spatial correlation indicates the correlation of the sensing data between nearby locations. Since environments are often smooth in a small area, and thus environmental values are similar at nearby locations. In this paper, we use the correlation coefficient to quantify this kind of correlation and dependence. Let $G_{(i)}$ denotes the *i*-th row of matrix *G*. In specific, $G_{(i)}$, $G_{(i')} \in R^n$ represent the data vectors of locations i and i'. The following metric $scf(i,i')$ (spatial correlation) between data at locations i and i' can be formulated as follows:

$$scf(i,i') = \frac{\sum_{j=1}^{n}(|G(i,j) - \bar{G}_{(i)}| \times |G(i',j) - \bar{G}_{(i')}|)}{\sqrt{\sum_{j=1}^{n}(G(i,j) - \bar{G}_{(i)})^2} \cdot \sqrt{\sum_{j=1}^{n}(G(i',j) - \bar{G}_{(i')})^2}} \tag{13}$$

where i and i' varies from 1 to m, $\bar{G}_{(i)} = \frac{1}{n}\sum_{j=1}^{n} G(i,j)$ and $\bar{G}_{(i')} = \frac{1}{n}\sum_{j=1}^{n} G(i',j)$. To avoid the existence of negative values in $scf(i,i')$, the absolute value function is added in the covariance function. Figure 6(c) plots the CDF of $scf(i,i')$, with the X-axis being the values of $scf(i,i')$ and Y-axis the cumulative probability. We find that the urban sensing data exhibits high spatial correlations in general.

In brief, the inherent features, i.e. low-rank feature, temporal stability feature, and spatial correlation feature discovered in urban sensing data allow us to perform spatial-temporal compressive sensing and quality assessment actions.

B. Cost map

In this paper, we estimated three different initial cost maps (i.e., i.i.d cost map, spatial correlated cost map and monotonic cost map) on the target datasets respectively. In the meantime, a dynamic, time-variant factor, i.e. the perception cost are considered in this paper. In our evaluation, we use $c_b = B^{1-b}$ (B = 2, $b \in [0,1]$) to denote the dynamic cost. The example of three different cost maps is given in Fig. 4. More specifically, the summary statistics over the three cost maps are shown in Fig. 7 (the unit of cost is CNY). As we can see in the violin diagram, each dot represents a sample cost and the height of the violin outline indicates the range of costs. Note that the range and std. deviation in CT1 are maximal compared to those in the other two cost maps while the mean and the minimal value of CT1 is smallest, that is to say, CT1 has more sample cost with small values. This fact can explain why our proposed algorithms will select more cells in CT1 to sense, and more details can be referred to the experiment results section.

C. Baselines

Since this paper addresses the practical sensing problem (usually a nonlinear system) with less historical monitoring data, we compare our cell selection strategies with two baselines: SIMP-GREEDY and QBC.

SIMP-GREEDY: Since there is typically a conflict between the informativeness and sample cost in a cell, the most straightforward strategy is to simply divide the informativeness by the sample cost. Thus, we can have the selection strategy as:

$$\arg\max_{\upsilon \in \mathcal{V} \setminus S_j} \ f_1(\upsilon)/f_2(\upsilon) \tag{14}$$

This strategy transforms a bi-objective problem into maximizing the single objective $f_1(\upsilon)/f_2(\upsilon)$ in each selection, which provides a simple solution for cost-quality beneficial selection, but it may fail when one of the two factors dominates the other [35]. Hence, SIMP-GREEDY is considered as a baseline.

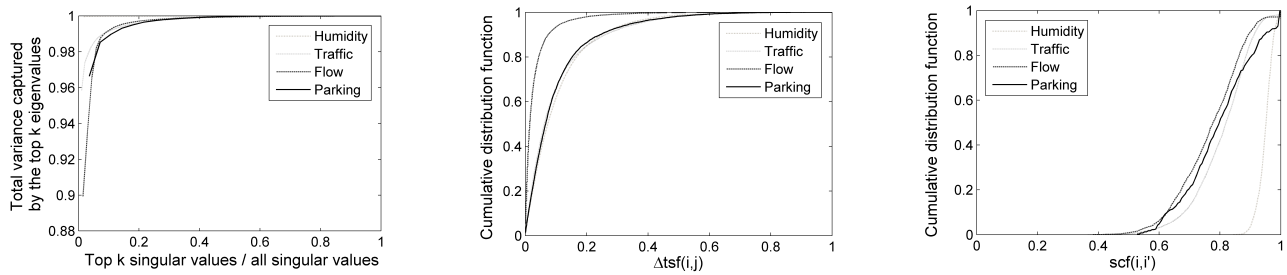
QBC: Previous works [13], [27] have proven the feasibility and satisfying performance of QBC of cell selection in Sparse MCS applications. Some "committee members" are contained in QBC to determine which salient cell to sense in the next task. More specifically, the "committee" is formed by different data inference algorithms, such as spatial-temporal compressive sensing, compressive sensing, K-Nearest Neighbors and SVD. Finally, it chooses the cell where the inferred data of various algorithms has the largest variance as the next selection for sensing without considering the cost-diversity. In other words, QBC tries to minimize the total costs of selecting cells by selecting the unsensed cells with the largest variance. Therefore, QBC is suitable as a baseline.

D. Experiment results

1) Errors of inferred value: We first compare the average inference error, i.e. MAE brought by different cell selection strategies while changing the number of selected cells for each cycle without considering (ϵ, p)-quality. As exhibited in Fig. 8, similar tendencies are observed over four types of sensing tasks. As the increment of the number of selected cells in each sensing cycle, the average inference errors drop rapidly. The fact implies that more information brought by the increasing selected cells promotes the accuracy of data inference. Note that the information modeling of our proposed strategies, i.e. POS and GCB-GREEDY and the baseline SIMP-GREEDY are based on QBC, and thus they share the comparable error levels theoretically. This fact is also confirmed by the experimental results though the inference error of our proposed strategies is better than that of the baselines in many circumstances. Next, we will evaluate and discuss the performances of our cell selection strategies by considering (ϵ, p)-quality, which is more practical in real-world applications.

2) The number and total costs of selected cells: Then we focus on analyzing the research objective – how much sample costs could our proposed algorithms save while further obtaining more informativeness to reduce the inference errors? The proposed strategies are compared to the baselines from three aspects: costs, selected cells and inference errors on four real-life datasets.

On the Parking occupancy rate sensing, for the predefined (ϵ, p)-quality, we set the error bound ϵ as 0.1 and p as 0.9 or 0.95. In other words, we require the inference error smaller than 0.1 for around 90% or 95% of cycles. The average number of selected cells for each cycle is shown in Fig. 9(a), where the baseline QBC always selects the fewest cells on three



(a) The low-rank feature revealed by the top k singular values

(b) The temporal stability feature revealed by differences between two consecutive intervals

(c) The spatial correlation feature revealed by the correlation between two subareas

Fig. 6. Inherent features in different datasets

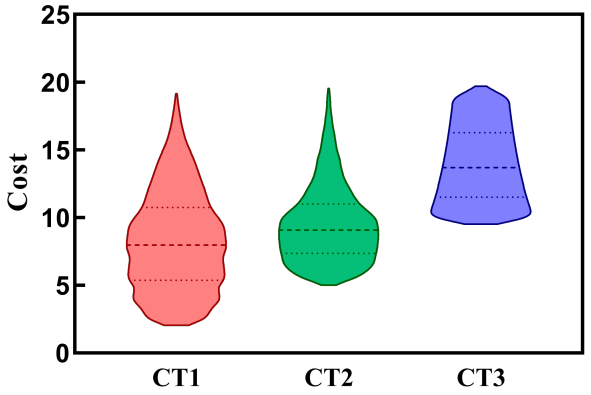


Fig. 7. The violin diagram of different cost maps. CT1, CT2, and CT3 denote the i.i.d with dynamic cost map, monotonic with dynamic cost map, and spatial correlated cost map, respectively; the three doted lines mark the 25% percentile, median, 75% percentile value in each violin plot, respectively

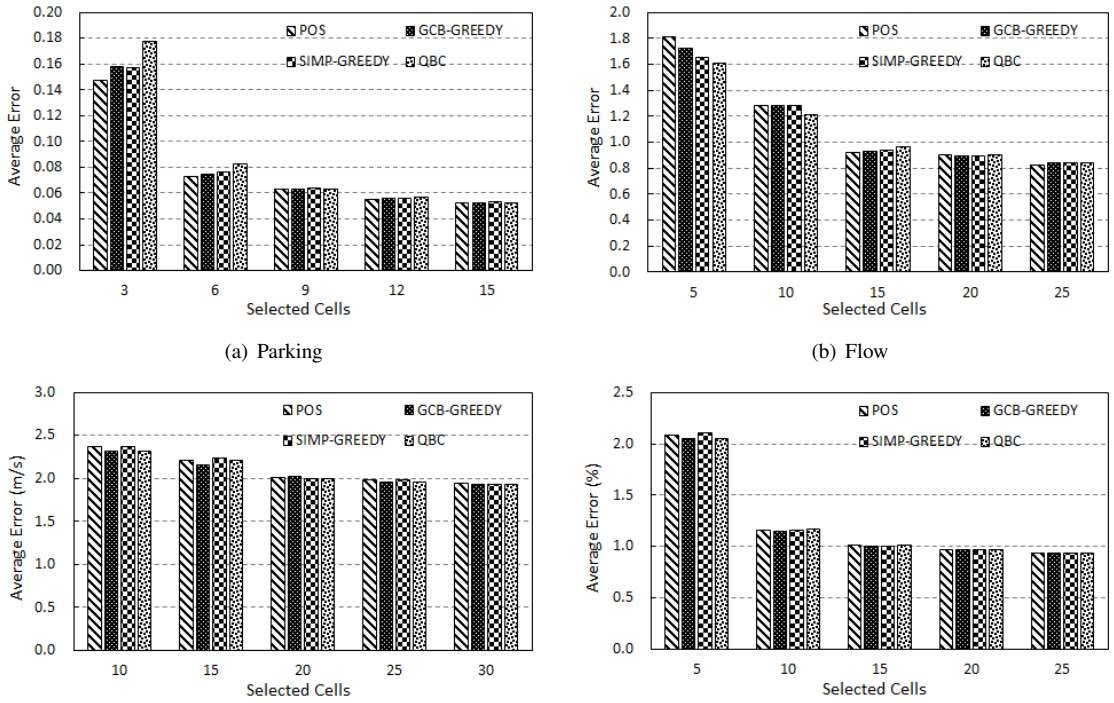
different cost maps, while GCB-GREEDY and POS can select 0.5%-4.2% (on average 2.8%) and 0.6%-5.2% (on average 3.5%) more subareas than QBC, respectively. Except for the circumstance of CT1 (95%), SIMP-GREEDY also selects a bit more cells (0.21%-0.9%, on average 0.5%). Note that in CT1 (i.i.d with dynamics cost map), our proposed strategies select more cells. The phenomenon can be explained by the statistics of cost maps since CT1 has more sample cost with small values. So our proposed algorithms may choose more than one cell in a selection (revealed in Fig. 3). In general, GCB-GREEDY and POS only need to select on average 6.67 (7.23), 6.74 (7.26) out of 30 cells for each sensing cycle to ensure the inference error below 0.1 in 90% (95%) of cycles, respectively. Though more cells are selected by our proposed strategies, the total costs of our proposed strategies outperform those of the baselines. Generally, QBC costs the most while POS saves the most costs, as shown in Fig. 9(b). Specifically, GCB-GREEDY and POS spend 1.6%-9.1% (on average 4.7%), 2.1%-11.2% (on average 5.7%) fewer costs than QBC. Meanwhile, our proposed strategies perform better than SIMP-GREEDY on cost saving in practically all circumstances. Especially in CT1, our proposed algorithms can achieve the best performance. Due to the simple greedy heuristic, SIMP-GREEDY cannot ensure a full superiority over QBC. Note that in the case of CT3 (90%), it even spends more cost than QBC. Subsequently, let us compare the inference errors. As shown in Appendix (Table IV), since our proposed strategies cover more subareas in each sensing cycle, which can provide more information for data inference, and thus enhance data accuracy compared to the baselines.

For the Flow and Traffic dataset, we observe a similar tendency in Fig. 9(c), (d), (e), and (f). It is noteworthy that our proposed strategies achieve better performance than the baselines since they leverage a more complex mechanism to balance the sample cost and information. Specifically, POS and GCB-GREEDY select more cells and save more costs compared to those in parking sensing tasks since the average number of selected cells in a time cycle becomes larger. Also, the inference error of our proposed strategies is obviously reduced.

On the Humidity dataset, our proposed strategies POS (GCB-GREEDY) can reduce inference errors by 6.1% to 10.1% (5.7% to 8.5%) compared with QBC, and 0.8% to 2.7% (0.2% to 2.2%) compared with SIMP-GREEDY, shown in Appendix (Table IV). Also, see results in Fig. 9(h), our proposed strategies POS (GCB-GREEDY) reduces the sample cost by 1.8% to 15.2% (1.4% to 15.02%) compared with QBC, and 1.0% to 8.5% (0.6% to 7.4%) compared with SIMP-GREEDY.

From the above-analyzed results, our proposed strategies undoubtedly outperform the baselines on the performance of decreasing inference errors and sample costs. Now we define a new indicator – Cost per cell (CPC) to see which strategy performs best. The results are shown in Fig. 10. On the Parking sensing task, the CPC of POS performs the best in all cases. Similarly, on the Flow, Traffic and Humidity dataset, POS also shows its advantage in all circumstances over other strategies. Thus, we can conclude POS is the best strategy considering the results.

Finally, let us see whether all the strategies can achieve the predefined task quality requirement. As shown in Appendix (Table V), most of the values are larger than its predefined p (all the methods adopt *LOO-SA* as the stopping criterion), and this result indicates that both our proposed strategies and the baselines can well satisfy the predefined quality requirement for most of the time. At meantime, we also observe that some



(a) Parking (b) Flow

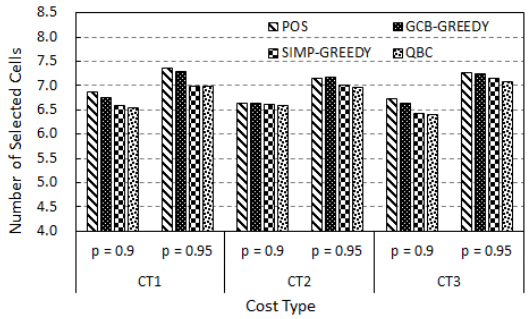
(c) Traffic (d) Humidity

Fig. 8. Average inference error of different sensing datasets under the condition of fixed number of selected cells for each cycle without considering (ϵ, p)-quality. The X-axis denotes the fixed number of selected cells while the Y-axis represents the corresponding average inference errors. As shown in this figure, the proposed strategies and baselines share the similar error levels since we do not leverage the quality assessment module and these strategies are all based on the same information modeling methods

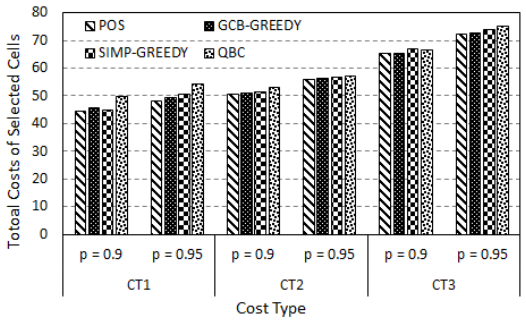
results are slightly less than the predefined p, for instance, $0.8992 < 0.9$ and $0.9496 < 0.95$, but the gap is quite small and acceptable. This is probably due to the fact that compressive sensing and Bayesian inference in our algorithms have the intrinsic probabilistic characteristics and would cause some minor errors. To ensure the accuracy of the results, each experiment sample was run five times. If time permits, more runs should be considered to avoid this probabilistic characteristic.

3) Results of different cost budgets: Further, to study how the change of cost budget in a selection will influence the evaluation results, we take the humidity sensing on CT1 as an example and conduct more experiments on POS strategy since POS has exhibited its superiority over other strategies. In the previous experiments, the cost budget in a selection is set to $B_{cost}^{one} = \max(f_2(\mathcal{V}_j))$, that is to say, B_{cost}^{one} equals the maximal sample cost in cycle j. And the reason for this setting is that cell selection strategies can consider and select any possible candidate subareas. Now we vary the ratio of cost budget to the maximal sample cost, and the results are shown in Fig. 11. Generally, when the ratio rises, more subareas are selected and the total sample costs correspondingly increase. This may because when the ratio is less than 1, the POS strategy omits some subareas with high sample costs; while when the ratio is greater than 1, the POS strategy has a greater cost budget to select more cells. Note that though the total sample costs are reduced in the low-cost budget scenarios, the inference errors increase significantly. When the ratio is less than 0.7, the results cannot even meet the predefined quality requirement. Thus, if a governor cares more about the cost reduction, the cost budget can be set slightly below the maximum sample cost. But to ensure the inference performance, we suggest the ratio should be set greater than 0.9.

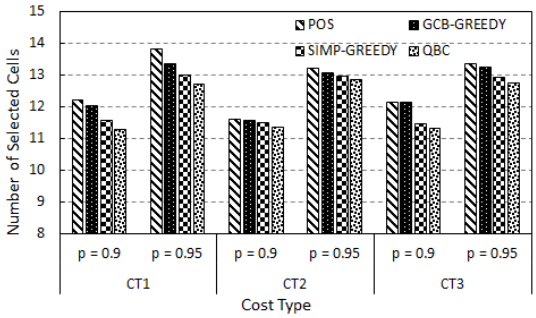
4) Results of leave some percentage out (LSPO) cross validation: Though the number of selected cells in most cycles is much smaller than the total number of cells, there still remains the situation that in some cycles more than a half or two thirds of the total cells are chosen since the data in these cycles is not necessarily accurately sensed due to the failure of sensors. In matrix completion, there is always a threshold of the observation rate, beyond which the performance should be satisfiable. Thus, in the above-mentioned situation, the number of observed entries (measured cells) is likely to go beyond the threshold, and consequently the performance may always be satisfiable by using the *LOO* evaluation, because leaving only one entry out may be too few and cannot show when the method can work and cannot work. In [52], [16], some other cross validation approaches are leveraged as the quality assessment method, such as *K*-fold cross validation and *leave-P-out* (*LPO*) cross validation. Note that *LOO* is a particular case of *K*-fold and *LPO*. In this paper, we conduct the evaluation by leaving various percentages (e.g. 10%, 20%, and 30%) of data out. Suppose that we have sensed m' cells out of all the m cells, the idea of *leave some percentage out* (*LSPO*) is that for each time, we leave some percentage, e.g. 10%, of the m' observations out (i.e. leave P observations



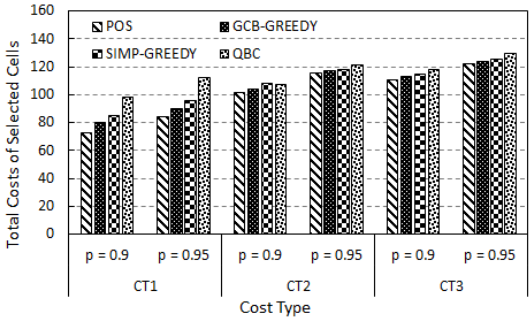
(a) Number of selected cells (Parking)



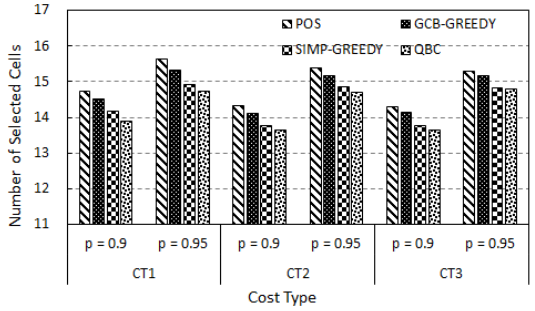
(b) Total costs (Parking)



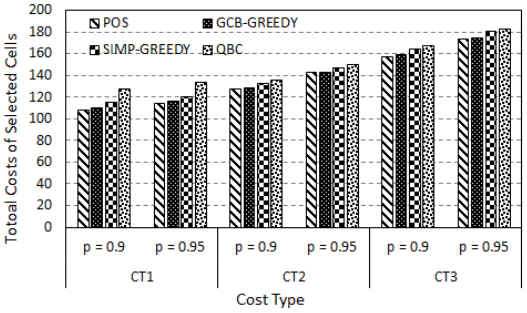
(c) Number of selected cells (Flow)



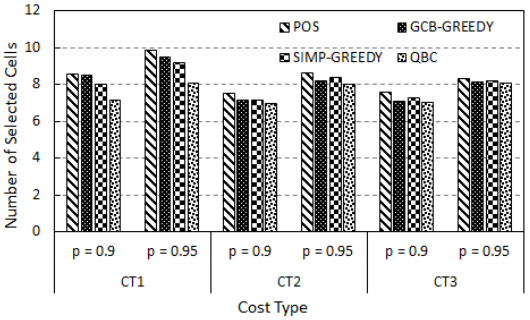
(d) Total costs (Flow)



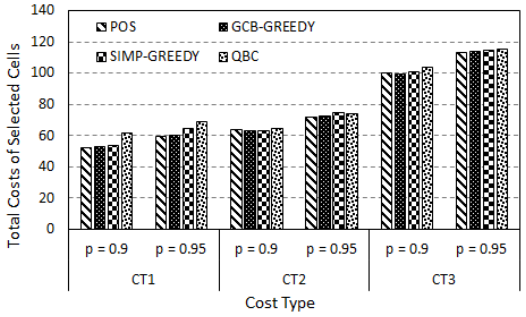
(e) Number of selected cells (Traffic)



(f) Total costs (Traffic)



(g) Number of selected cells (Humidity)

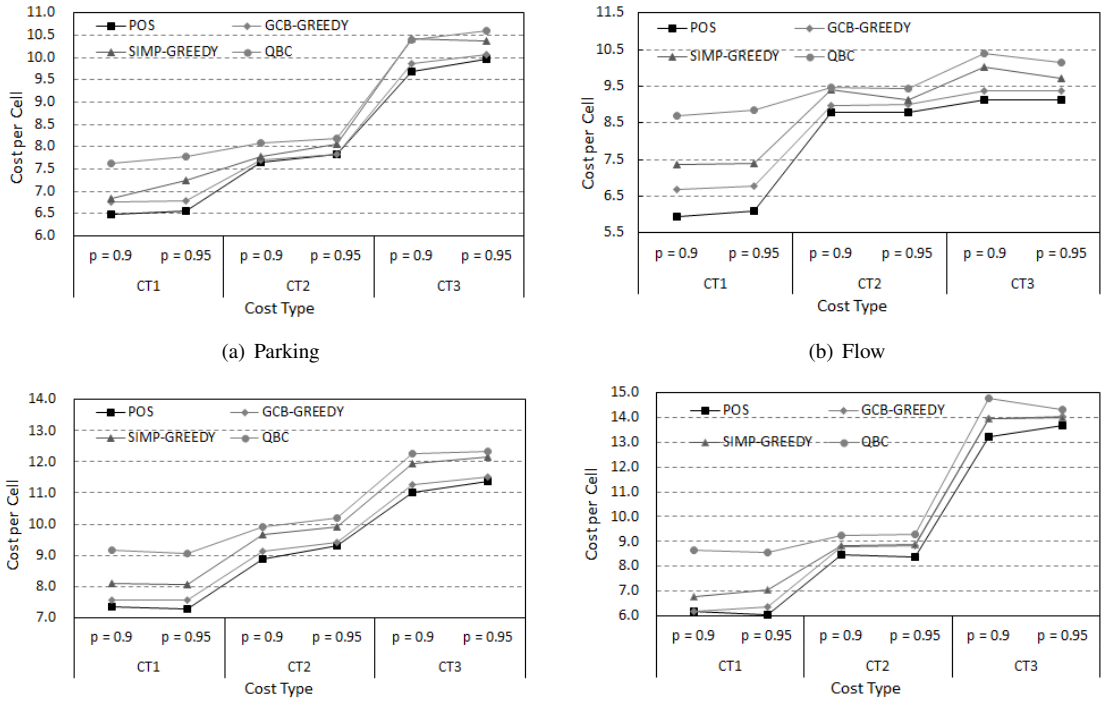


(h) Total costs (Humidity)

Fig. 9. The number and total costs of selected cells for Parking, Flow, Traffic and Humidity under the condition of considering (ϵ, p)-quality. The X-axis denotes different types of cost map; the Y-axis represents the number of selected cells in (a), (c), (e), (g) while the Y-axis represents total costs of selected cells in (b), (d), (f), (h)

out, P equals $ceil(m' \times percentage)$) and infer them based on the rest (m'-P) observations. Here we take the humidity sensing task as an example, set the error bound ϵ as 1.5% and p as 0.9 and experiment on percentages 10%, 20%, 30%, 40% and 50%, respectively. Each experiment sample is repeated 5 times (the results are averaged), and the corresponding results are shown in Fig. 12 and Appendix (Table. VI).

It can be concluded that compared to *LOO*, leveraging *LSPO* incurs a bit more selected cells as well as sensing costs, and also improves the inference results to some extent. Specifically, with the increment of the percentage, the average number of



(a) Parking

(b) Flow

(c) Traffic

(d) Humidity

Fig. 10. Cost per cell of different cell selection strategies for Parking, Flow, Traffic and Humidity sensing tasks

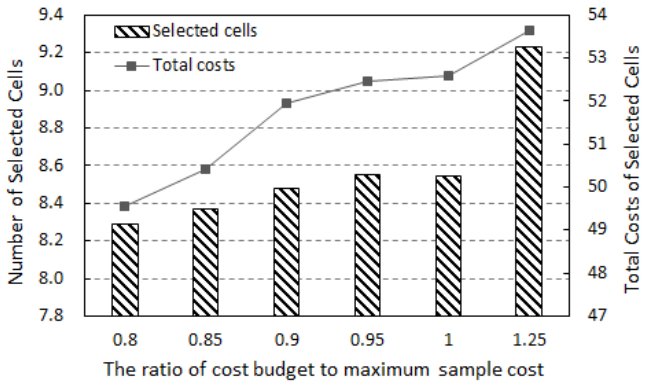


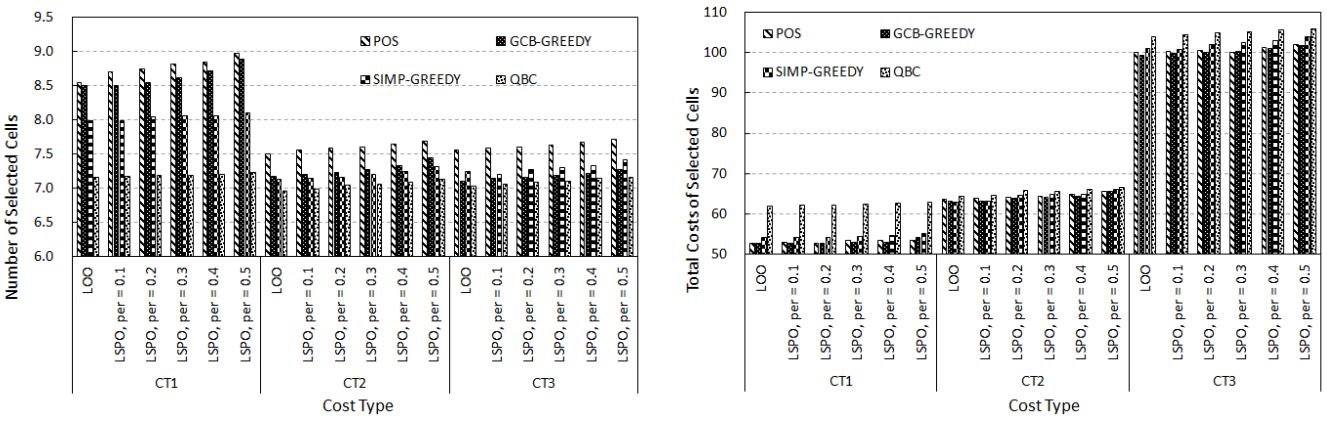
Fig. 11. Results of different ratios of cost budget to maximum sample cost

selected cells in each cycle increases slightly. This is because leaving more cells out in the above-mentioned situation would reduce the observation rate and require more cells to be sensed. In most cases, the corresponding sensing costs also increase slightly, and the inference error in unsensed cells decreases a little. The findings reveal that *LSPO* can better handle with the inaccurate sensing situation in some cycles compared to *LOO* and avoid the invalid operation when the observation rate exceeds the threshold in matrix completion. But *LSPO* cross validation requires to learn and validate $C_{m'}^{p}$ times, so as the value of m' becomes too big, it would be impossible to calculate. Therefore, in terms of computational efficiency, *LOO* is a good choice; and considering the reduction of inference error and the problem of coping with the observation rate over threshold, *LSPO* is a better choice.

5) Running time: Finally, since the baseline QBC has demonstrated its feasibility of running time performance in the real-life scenario, we report the computation time of our proposed strategies and SIMP-GREEDY to see whether they can also satisfy the runtime requirements. We run the experiments on a desktop computer (Intel Core i7-8559U CPU @ 2.70GHz, 16GB RAM, Windows 10) with Python3.7. Table III lists the running time for different stages of the whole task assignment process. As we can see in Table III, the 'Quality Assessment' module costs the most since it needs to run the 'Data inference' module for many rounds to judge whether the sensing cycle can stop or continue. Though our proposed strategies use QBC as the basis to estimate the information in unsensed cells, the computation time of GCB-GREEDY is even reduced. This is because the GCB-greedy algorithm can ensure a fixed runtime and may select more than one cell in a selection. Despite POS is the most time consuming, the total runtime for allocating a new task is no more than 15.4 seconds (i.e., estimating the task quality once and, if it cannot meet the predefined (ϵ, p)-quality, finds the next sensing cell). Therefore, we believe the efficiency of our proposed methods can also satisfy most real-world applications.

TABLE III
RUNNING TIME FOR EACH STAGE

	Parking	Flow	Traffic	Humidity
Cell Selection – QBC	0.91s	1.19s	1.36s	1.15s
Cell Selection – SIM-GREEDY	1.26s	1.65s	1.90s	1.61s
Cell Selection – GCB-GREEDY	0.67s	1.13s	1.27s	1.04s
Cell Selection – POS	1.68s	2.74s	3.21s	2.53s
Data Inference	0.51s	0.89s	1.27s	0.75
Quality Assessment	< 3.9s	< 8.6s	< 12.1s	< 7.2s



(a) Number of selected cells

(b) Total costs

Fig. 12. Results of leaving different percentages out for Humidity sensing tasks

6) Discussion: In this subsection, we will conclude the experimental phenomena and discuss some drawbacks of this paper.

Our proposed cell selection method explicitly outperform the baselines, with two strategies (Pareto optimization and generalized cost-benefit greedy) from three aspects: less sample cost, more selected cells with sensing values, and less inference error, since we leverage a complex mechanism to minimize the total cost and maximize the informativeness. In other words, we select more subareas on the premise of reducing or not increasing the overall cost. Since more cells are sensed with real measurements, the inference error is also improved. Whatever the sample cost type and sensing task type is, POS strategy achieves the best performance; however, if the running time is considered, GCB-GREEDY is a comparable strategy. *Experimental results demonstrate the feasibility of the proposed cost-quality beneficial cell selection method.*

Compared to the results in monotonic cost map (CT2) and spatial correlated cost map (CT3), POS strategy and GCB-GREEDY strategy perform much better in sample cost reduction and inference error decline under i.i.d. cost map (CT1). It is because CT1 owns cost values with a bigger range and std. deviation and has more sample cost with small values, thus it provides Pareto optimization and generalized cost-benefit greedy more chances to select more than one subarea in a selection. Thus, the average number of selected cells in CT1 of Pareto optimization and generalized cost-benefit greedy is greater than that in CT2 and CT3. More selected cells mean more information for recovery, and thus the inference error reduction in CT1 is better. *This finding implicates that our proposed framework and cell selection strategies is able to handle various kinds of cost inconstancy, especially when the cost map has a bigger range and standard deviation.*

However, there remain some drawbacks in the present work. Firstly, we still leverage *LOO-SA* as the stopping criterion, in which the practical relationship between statistical results and the stop condition is simplified. Thus, when the observed entries are beyond the threshold in a cycle, the performance may always be satisfiable. So that it is unclear whether *LOO-SA* is working or not in this situation. Instead, *LSPO-SA* would be a good choice for quality assessment. Secondly, since we use the Query by Committee method to estimate the uncertainty in each unsensed cells. But the direct relationship between the uncertainty and the quality of reconstruction remains to be proven. Finally, the accuracy of data acquisition influences the overall performance of Sparse MCS to some extent, which needs further discussion and analysis.

In a few cases, we observe that naive greedy strategy even performs a bit better than the generalized cost-benefit greedy strategy. It may because compressive sensing and Bayesian inference in our algorithms have the intrinsic probabilistic characteristics and would cause some minor errors. If time permits, more runs should be considered to avoid this probabilistic characteristic. It may also because there exist certain measurement errors in the raw data, which affects the performance of our proposed strategies in some cases.

The practicality of this work may be limited on the simulation-based results without real-world applications and practical experiments. The issue of human factor is avoided by a perfect participant assumption in this paper. However, a participant may fail, deny, or be late in doing the assigned task. The probability of the failure should be different participant by participant, since their personality is different. Thus, we would clarify how the proposed methods handle human factors happening in the real-world applications by conducting real experiments in the future.

VII. Conclusion

Crowdsensing, as a typical way of urban computing, has shown its advantage in pervasive sensing and knowledge discovery. However, sample costs of high-quality data still hamper MCS from being utilized at a large-scale. Thus, in this paper, we incorporate cost-diversity into the cell selection process. To that end, a novel three-step cell selection approach (information modeling, cost estimation, and cost-quality beneficial selection) is proposed with the target of minimizing the total sample costs and maximizing the beneficial informativeness in the selected cells (further reducing the error of inferred results). After reasonable approximation of the cost and discussion on the properties of the optimization goals,

we propose two selection strategies, namely GCB-GREEDY and POS, to solve the optimization model. We evaluate the proposed strategies by comparing them to two baselines, i.e. QBC and SIMP-GREEDY, on four real-life sensing datasets and three different cost maps. Results explicitly demonstrate the effectiveness and applicability of our strategies in real-world MCS systems. Our strategies can save sample costs and reduce inference errors in the meantime. Also, we analyze the potential of applying our proposed cost-quality beneficial crowd-powered way in real-life urban-scale sensing and actuation.

In the future, we will continue to improve this work from the following aspects. We will firstly incorporate the sample cost diversity into the deep reinforcement learning-based cell selection method. Secondly, we will improve the insufficient participant scenario by considering user's historical mobility traces where relocate participants to new task locations. Last but not the least, we will provide a mathematical proof of the relationship between the informativeness of a cell and the reconstruction performance in the nonlinear system scenarios.

VIII. Appendix

Three tables in section of Evaluation Performance are listed here.

TABLE IV
AVERAGE INFERENCE ERRORS OF DIFFERENT CELL SELECTION STRATEGIES UNDER THE CORRESPONDING QUALITY REQUIREMENT

Parking					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(0.1, 0.90)	0.0697	0.0712	0.0747	0.0818
	(0.1, 0.95)	0.0669	0.0674	0.0706	0.0794
CT2	(0.1, 0.90)	0.0701	0.0719	0.0732	0.0820
	(0.1, 0.95)	0.0682	0.0683	0.0719	0.0806
CT3	(0.1, 0.90)	0.0702	0.0894	0.0741	0.0822
	(0.1, 0.95)	0.0675	0.0685	0.0727	0.0811
Flow					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(1.2, 0.90)	0.9321	0.9820	1.0328	1.0544
	(1.2, 0.95)	0.9185	0.9377	0.9687	0.9866
CT2	(1.2, 0.90)	1.0029	1.0058	1.0057	1.0070
	(1.2, 0.95)	0.9239	0.9365	0.9771	0.9800
CT3	(1.2, 0.90)	0.9554	0.9504	1.0305	1.0261
	(1.2, 0.95)	0.9293	0.9309	0.9747	0.9851
Traffic					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(2.5m/s, 0.90)	2.2456	2.2706	2.2904	2.3154
	(2.5m/s, 0.95)	2.1924	2.2127	2.2269	2.2504
CT2	(2.5m/s, 0.90)	2.2778	2.2822	2.3068	2.3177
	(2.5m/s, 0.95)	2.2045	2.2178	2.2566	2.2666
CT3	(2.5m/s, 0.90)	2.2754	2.2839	2.3108	2.3289
	(2.5m/s, 0.95)	2.2121	2.2202	2.2556	2.2645
Humidity					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(1.5%, 0.90)	1.3244	1.3444	1.3753	1.4697
	(1.5%, 0.95)	1.2694	1.2988	1.3049	1.4122
CT2	(1.5%, 0.90)	1.3604	1.3732	1.3865	1.4655
	(1.5%, 0.95)	1.3301	1.3364	1.3397	1.4322
CT3	(1.5%, 0.90)	1.3733	1.3757	1.3834	1.4604
	(1.5%, 0.95)	1.3170	1.3249	1.3269	1.4284

TABLE V
FACTION OF THE CYCLES WHOSE ERRORS ARE LOWER THAN THE CORRESPONDING QUALITY REQUIREMENT

Parking					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(0.1, 0.90)	0.9084	0.9049	0.9026	0.9003
	(0.1, 0.95)	0.9553	0.9542	0.9530	0.9519
CT2	(0.1, 0.90)	0.9061	0.9003	0.8992	0.8995
	(0.1, 0.95)	0.9507	0.9537	0.9488	0.9530
CT3	(0.1, 0.90)	0.9107	0.8981	0.9003	0.9018
	(0.1, 0.95)	0.9530	0.9496	0.9542	0.9496
Flow					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(1.2, 0.90)	0.9341	0.9267	0.9194	0.9158
	(1.2, 0.95)	0.9780	0.9780	0.9707	0.9634
CT2	(1.2, 0.90)	0.9231	0.9084	0.9158	0.9085
	(1.2, 0.95)	0.9560	0.9597	0.9560	0.9524
CT3	(1.2, 0.90)	0.9304	0.9267	0.9121	0.9121
	(1.2, 0.95)	0.9524	0.9670	0.9667	0.9634
Traffic					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(2.5m/s, 0.90)	0.9121	0.9231	0.9048	0.9011
	(2.5m/s, 0.95)	0.9707	0.9707	0.9634	0.9597
CT2	(2.5m/s, 0.90)	0.9231	0.9084	0.9011	0.8974
	(2.5m/s, 0.95)	0.9560	0.9634	0.9597	0.9524
CT3	(2.5m/s, 0.90)	0.9048	0.9194	0.8974	0.9048
	(2.5m/s, 0.95)	0.9597	0.9780	0.9524	0.9451
Humidity					
	(ϵ, p)	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	(1.5%, 0.90)	0.9101	0.9109	0.9094	0.8920
	(1.5%, 0.95)	0.9524	0.9645	0.9630	0.9517
CT2	(1.5%, 0.90)	0.9026	0.9018	0.9079	0.8927
	(1.5%, 0.95)	0.9494	0.9630	0.9592	0.9456
CT3	(1.5%, 0.90)	0.8995	0.9063	0.9086	0.8995
	(1.5%, 0.95)	0.9502	0.9502	0.9464	0.9479

TABLE VI
AVERAGE INFERENCE ERRORS OF DIFFERENT QUALITY ASSESSMENT METHODS ON HUMIDITY SENSING TASK UNDER $(\epsilon = 1.5, p = 0.9)$ QUALITY REQUIREMENT

Humidity					
	Quality Assessment	POS	GCB-GREEDY	SIMP-GREEDY	QBC
CT1	*LOO*	1.3244	1.3444	1.3753	1.4697
	LSPO, per=0.1	1.3243	1.3417	1.3747	1.4698
	LSPO, per=0.2	1.3237	1.3403	1.3742	1.4696
	LSPO, per=0.3	1.3219	1.3392	1.3732	1.4692
	LSPO, per=0.4	1.3166	1.3386	1.3720	1.4688
	LSPO, per=0.5	1.3155	1.3375	1.3717	1.4680
CT2	*LOO*	1.3604	1.3732	1.3865	1.4655
	LSPO, per=0.1	1.3604	1.3733	1.3859	1.4657
	LSPO, per=0.2	1.3599	1.3724	1.3847	1.4654
	LSPO, per=0.3	1.3591	1.3721	1.3842	1.4648
	LSPO, per=0.4	1.3582	1.3716	1.3839	1.4637
	LSPO, per=0.5	1.3566	1.3705	1.3834	1.4631
CT3	*LOO*	1.3733	1.3757	1.3834	1.4604
	LSPO, per=0.1	1.3731	1.3756	1.3836	1.4601
	LSPO, per=0.2	1.3723	1.3748	1.3814	1.4593
	LSPO, per=0.3	1.3717	1.3739	1.3812	1.4582
	LSPO, per=0.4	1.3708	1.3732	1.3807	1.4580
	LSPO, per=0.5	1.3702	1.3729	1.3804	1.4573

Acknowledgment

This study is supported by the National Natural Science Foundation of China under Grant Nos.71673292, 21808181, 61673388, 71673294 and National Social Science Foundation of China under Grant No.17CGL047. This research is also partially supported by the EU Horizon 2020 research and innovation program under grant agreements No. 825134 (ARTICONF), No. 824068 (ENVRI-FAIR), and No. 862409 (BLUECLOUD).

References

[1] H. Sun, W. Shi, X. Liang, and Y. Yu, "Vu: Edge computing-enabled video usefulness detection and its application in large-scale video surveillance systems," *IEEE Internet of Things Journal*, 2019.

[2] J. Xia, G. Cheng, S. Gu, and D. Guo, "Secure and trust-oriented edge storage for internet of things," *IEEE Internet of Things Journal*, 2019.

[3] S. Devarakonda, P. Sevusu, H. Liu, R. Liu, L. Iftode, and B. Nath, "Real-time air quality monitoring through mobile sensing in metropolitan areas," in *Proceedings of the 2nd ACM SIGKDD international workshop on urban computing*, 2013, pp. 1–8.

[4] E. D'Hondt, J. Zaman, E. Philips, E. G. Boix, and W. De Meuter, "Orchestration support for participatory sensing campaigns," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2014, pp. 727–738.

[5] Z. Liu, S. Jiang, P. Zhou, and M. Li, "A participatory urban traffic monitoring system: the power of bus riders," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2851–2864, 2017.

[6] H. Aly, A. Basalamah, and M. Youssef, "Automatic rich map semantics identification through smartphone-based crowd-sensing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2712–2725, 2016.

[7] J. Wang, L. Wang, Y. Wang, D. Zhang, and L. Kong, "Task allocation in mobile crowd sensing: State-of-the-art and future opportunities," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3747–3757, 2018.

[8] J. Wang, Y. Wang, D. Zhang, F. Wang, H. Xiong, C. Chen, Q. Lv, and Z. Qiu, "Multi-task allocation in mobile crowd sensing with individual task quality assurance," *IEEE Transactions on Mobile Computing*, vol. 17, no. 9, pp. 2101–2113, 2018.

[9] M. H. Cheung, R. Southwell, F. Hou, and J. Huang, "Distributed time-sensitive task selection in mobile crowdsensing," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2015, pp. 157–166.

[10] W. Liu, Y. Yang, E. Wang, Z. Han, and X. Wang, "Prediction based user selection in time-sensitive mobile crowdsensing," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2017, pp. 1–9.

[11] 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, 2018.

[12] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for wsns: Sparse signal modeling and monitoring framework," *IEEE Transactions on Wireless Communications*, vol. 11, no. 10, pp. 3447–3461, 2012.

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

[14] S. Chakraborty, J. Zhou, V. Balasubramanian, S. Panchanathan, I. Davidson, and J. Ye, "Active matrix completion," in *2013 IEEE 13th International Conference on Data Mining*. IEEE, 2013, pp. 81–90.

[15] W. Liu, L. Wang, E. Wang, Y. Yang, D. Zeghlache, and D. Zhang, "Reinforcement learning-based cell selection in sparse mobile crowdsensing," *Computer Networks*, vol. 161, pp. 102–114, 2019.

[16] W. Liu, Y. Yang, E. Wang, L. Wang, D. Zeghlache, and D. Zhang, "Multi-dimensional urban sensing in sparse mobile crowdsensing," *IEEE Access*, vol. 7, pp. 82 066–82 079, 2019.

[17] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, and D. Zhang, "Active sparse mobile crowd sensing based on matrix completion," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 195–210.

[18] L. Xu, X. Hao, N. D. Lane, X. Liu, and T. Moscibroda, "Cost-aware compressive sensing for networked sensing systems," in *Proceedings of the 14th international conference on Information Processing in Sensor Networks*, 2015, pp. 130–141.

[19] J. Liu, H. Shen, H. S. Narman, W. Chung, and Z. Lin, "A survey of mobile crowdsensing techniques: A critical component for the internet of things," *ACM Transactions on Cyber-Physical Systems*, vol. 2, no. 3, pp. 1–26, 2018.

[20] L. Xu, X. Hao, N. D. Lane, X. Liu, and T. Moscibroda, "More with less: Lowering user burden in mobile crowdsourcing through compressive sensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015, pp. 659–670.

[21] S. He and K. G. Shin, "Steering crowdsourced signal map construction via bayesian compressive sensing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1016–1024.

[22] C. Meng, H. Xiao, L. Su, and Y. Cheng, "Tackling the redundancy and sparsity in crowd sensing applications," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, 2016, pp. 150–163.

[23] X. Xia, Y. Zhou, J. Li, and R. Yu, "Quality-aware sparse data collection in mec-enhanced mobile crowdsensing systems," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1051–1062, 2019.

[24] Z. Liu, Z. Li, and K. Wu, "Unitask: A unified task assignment design for mobile crowdsourcing-based urban sensing," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6629–6641, 2019.

[25] X. Zhang, Z. Yang, and Y. Liu, "Vehicle-based bi-objective crowdsourcing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3420–3428, 2018.

[26] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1654–1662.

[27] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "Ccs-ta: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proceedings of the 2015 ACM international joint conference on pervasive and ubiquitous computing*, 2015, pp. 683–694.

[28] L. Wang, D. Zhang, D. Yang, A. Pathak, C. Chen, X. Han, H. Xiong, and Y. Wang, "Space-ta: Cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 2, pp. 1–28, 2017.

[29] L. Wang, D. Zhang, D. Yang, B. Y. Lim, and X. Ma, "Differential location privacy for sparse mobile crowdsensing," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 2016, pp. 1257–1262.

[30] L. Wang, W. Liu, D. Zhang, Y. Wang, E. Wang, and Y. Yang, "Cell selection with deep reinforcement learning in sparse mobile crowdsensing," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1543–1546.

[31] W. Liu, Y. Yang, E. Wang, and J. Wu, "User recruitment for enhancing data inference accuracy in sparse mobile crowdsensing," *IEEE Internet of Things Journal*, 2019.

[32] C. Qian, Y. Yu, and Z.-H. Zhou, "Subset selection by pareto optimization," in *Advances in Neural Information Processing Systems*, 2015, pp. 1774–1782.

[33] C. Qian, J.-C. Shi, Y. Yu, and K. Tang, "On subset selection with general cost constraints." in *IJCAI*, vol. 17, 2017, pp. 2613–2619.

[34] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[35] S.-J. Huang, M. Xu, M.-K. Xie, M. Sugiyama, G. Niu, and S. Chen, "Active feature acquisition with supervised matrix completion," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1571–1579.

[36] X. Hao, L. Xu, N. D. Lane, X. Liu, and T. Moscibroda, "Density-aware compressive crowdsensing," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2017, pp. 29–40.

[37] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, 2009, pp. 267–278.

[38] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 662–676, 2011.

[39] S. He and K. G. Shin, "Spatio-temporal adaptive pricing for balancing mobility-on-demand networks," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 4, pp. 1–28, 2019.

[40] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. CRC press, 2013.

[41] R. W. Johnson, "An introduction to the bootstrap," *Teaching Statistics*, vol. 23, no. 2, pp. 49–54, 2001.

[42] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," 2011.

[43] H. Zhang and Y. Vorobeychik, "Submodular optimization with routing constraints," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[44] C. Qian, Y. Yu, and Z.-H. Zhou, "An analysis on recombination in multi-objective evolutionary optimization," *Artificial Intelligence*, vol. 204, pp. 99–119, 2013.

[45] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: concepts, methodologies, and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 3, pp. 1–55, 2014.

[46] S. Jiang, G. A. Fiore, Y. Yang, J. Ferreira Jr, E. Frazzoli, and M. C. González, "A review of urban computing for mobile phone traces: current methods, challenges and opportunities," in *Proceedings of the*