RESEARCH-ARTICLE

Active Sparse Mobile Crowd Sensing Based on Matrix Completion

KUN XIE, Hunan University, Changsha, Hunan, China

XIAOCAN LI, Hunan University, Changsha, Hunan, China

XIN WANG, Stony Brook University, Stony Brook, NY, United States

GAOGANG XIE, University of Chinese Academy of Sciences, Beijing, China

JIGANG WEN, Institute of Computing Technology Chinese Academy of Sciences, Beijing, Beijing, China

DAFANG ZHANG, Hunan University, Changsha, Hunan, China

Published: 25 June 2019

SIGMOD/PODS '19: International Conference on Management of Data
June 30 - July 5, 2019
Amsterdam, Netherlands

Conference Sponsors:
SIGMOD

Active Sparse Mobile Crowd Sensing Based on Matrix Completion

Kun Xie*
College of Computer Science and Electronic Engineering, Hunan University
Changsha, Hunan, China
xiekun@hnu.edu.cn

Xiaocan Li†
College of Computer Science and Electronic Engineering, Hunan University
Changsha, Hunan, China
hnulxc@hnu.edu.cn

Xin Wang
Department of Electrical and Computer Engineering, State University of New York at Stony Brook
New York, USA
x.wang@stonybrook.edu

Gaogang Xie
State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, The School of Computer and Control Engineering, University of Chinese Academy of Sciences
Beijing, China
xie@ict.ac.cn

Jigang Wen
State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
wenjigang@ict.ac.cn

Dafang Zhang
College of Computer Science and Electronic Engineering, Hunan University
Changsha, Hunan, China
dfzhang@hnu.edu.cn

ABSTRACT

A major factor that prevents the large scale deployment of Mobile Crowd Sensing (MCS) is its sensing and communication cost. Given the spatio-temporal correlation among the environment monitoring data, matrix completion (MC) can be exploited to only monitor a small part of locations and time, and infer the remaining data. Rather than only taking random measurements following the basic MC theory, to further reduce the cost of MCS while ensuring the quality of missing data inference, we propose an Active Sparse MCS (AS-MCS) scheme which includes a bipartite-graph-based sensing scheduling scheme to actively determine the sampling positions in each upcoming time slot, and a bipartite-graph-based matrix completion algorithm to robustly and accurately recover the un-sampled data in the presence of sensing and communications errors. We also incorporate the sensing cost into the bipartite-graph to facilitate low cost sample selection and consider the incentives for MCS. We have conducted extensive performance studies using the data sets from the monitoring of PM 2.5 air condition and road traffic speed, respectively. Our results demonstrate that our AS-MCS scheme can recover the missing data at very high accuracy with the sampling ratio only around 11%, while the peer matrix completion algorithms with similar recovery performance requires up to 4-9 times the number of samples of ours for both the data sets.

KEYWORDS

Mobile Crowd Sensing (MCS), Matrix Completion

ACM Reference Format:
Kun Xie, Xiaocan Li, Xin Wang, Gaogang Xie, Jigang Wen, and Dafang Zhang. 2019. Active Sparse Mobile Crowd Sensing Based on Matrix Completion. In *2019 International Conference on Management of Data (SIGMOD '19), June 30-July 5, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3299869.3319856

*Corresponding author
†Corresponding author

SIGMOD '19, June 30-July 5, 2019, Amsterdam, Netherlands

ACM ISBN 978-1-4503-5643-5/19/06…$15.00
https://doi.org/10.1145/3299869.3319856

1 INTRODUCTION

The proliferation of smart phones contributes to the prosperity of a novel sensing paradigm called Crowd Sensing, where individuals with sensing and computing devices collectively measure and map phenomenon of common interest. In crowd sensing applications, humans can work as the sensor carriers or even the sensors to report the conditions of the surrounding environment, such as weather, traffic, air quality, etc. Due to the mobility of carriers, crowd sensing is also termed mobile crowd-sensing (MCS) [10] to refer to a broad range of community sensing paradigms.

Different from traditional wireless sensor networks (WSNs) [8, 9, 41]which usually leverage dedicated sensors to acquire real-world conditions, MCS utilizes citizens' off the-shelf smartphones to capture social and urban dynamics. Compared to WSNs, the grassroot participation offers MCS a number of advantages: (1) MCS leverages existing sensing and communication infrastructure, and therefore, its deployment cost is much lower than that under WSNs; (2) The inherent mobility of mobile users provides unprecedented spatiotemporal coverage compared to static sensor network deployments.

Many MCS applications require the long-term monitoring of a large area. For example, a city planning and management agency may be interested in the environment conditions of an urban area, which are tracked with a sensing map with the target area divided into N grids. The data monitoring over T time slots can be organized as a $X_{N\times T}$ matrix.

Ideally, the planner would like to get the complete sensing map. However, not all areas are covered by mobile users that are willing to provide the sensing data. The coverage ratio is often used as a quality metric to measure how many subareas (grid cells) of the target domain have been covered, and whether sufficient data have been collected from participants [2, 7, 11, 32, 46, 47, 50]. Existing research on task allocation in MCS primarily takes full coverage [32, 47] or coverage with a high probability as a constraint [2, 11, 46, 50]. As a result, an organizer has to collect at least one sample from each grid in a time slot or get data from most of the cells in the target area [46, 47], which would incur a large sensing and communication cost if the monitoring area is large and the data need to be collected over a long term. This will prevent the practical and large-scale deployment of MCS systems.

As almost all physical conditions monitored are continuous, sensory data generally exhibit strong spatio-temporal correlation [35], thus the environment matrix X has a low-rank feature. This provides the prerequisite for using the matrix completion (MC) [5, 16] to gather environment data to reduce the cost, where only a small set of grids need to be sampled and monitored by participants and the data of the remaining matrix entries can be inferred. Compared with the traditional converge-based task allocations, MC-based data gathering provides a new avenue for low cost MCS monitoring.

Some recent studies apply matrix completion to MCS [18, 28, 37, 38, 43, 45, 52], with the major focus on finding various algorithms to infer the missing values instead of sampling scheduling. Without considering whether the samples carry sufficient information for matrix completion, these techniques may suffer from big errors to infer the un-sampled data even though they can output their best estimation given a few known entries.

To guarantee the data quality, several theoretical contributions are proposed in matrix completion. For example, Candes and Recht [5] prove that an n_1 times n_2 matrix of rank r should take at least $m > Cn^{\frac{6}{5}} r \log(n)$ samples for the matrix completion to succeed with a high probability, where $n = max(n_1, n_2)$. Several follow-on studies develop slightly different sampling thresholds, but all are essentially $O(nr\log(n))$ [16, 29]. Derived based on the random sampling, these bounds are loose. Our recent studies show that the number of samples taken based on this bound is significantly larger than needed. Moreover, some recent results [6, 15, 34] show that different sampling locations would bring different information for the matrix recovery. If locations are well selected, a small number of samples would be able to bring sufficient information, which helps to reduce the sample number.

Rather than only taking random samples, to further reduce the cost of MCS while reliably obtaining the complete sensing map in presence of practical sensing and communication errors, we would like to investigate the possibility and techniques to perform an *active* sparse mobile crowd sensing where sample locations in the upcoming time slot can be carefully selected to provide more information for lower sensing cost and to reduce inference errors. With only the information of historical data, it is hard to determine how to schedule the future sampling to guarantee the performance of data recovery, and scheduling is made even hard with many practical factors to consider. Some challenging issues are:

- **How to determine the number of samples to take?** To achieve low cost measurement, redundant samples should be minimized. However, it is very hard to determine how many samples are sufficient to obtain the complete sensing map without knowing the information of the future data. There is also a need to consider the incentive and the total sensing cost when making the sample selection schedule.

- **How to select the locations to take samples?** Different sampling locations may bring different information to the matrix completion procedure, which further impacts the recovery performance. Without a prior knowledge of the future data, it is very difficult to select the sampling points. The locations chosen for sensing are also constrained by the geographic distributions of candidate participants. If there are not participants, its data can only be inferred. In addition, incentive is often provided in MCS [13], where the cost of sensing can be unbalanced in different geographical locations due to the difference in area coverage and density of participants. In this case, the selection of sample locations needs also to consider the total cost.
- **How to ensure the matrix recovery to be robust to noise and small errors?** In a practical MCS system, the data observed may include unavoidable noise or small errors during transmissions, which may have significant impact on the recovery performance. This requires carefully determining the samples and reliably inferring unsampled data.

To conquer the challenges, we propose an Active Sparse MCS scheme (AS-MCS) based on matrix completion to preselect sufficient samples with minimum sensing cost while accurately inferring the un-sampled monitoring data. Specially, we develop a scheduling algorithm to carefully select the grid locations to take samples in each upcoming time slot based on data already taken, taking into account the difference in information, cost for the sensing of different grid locations and the impact of the samples on the reliability of solutions. As no existing studies investigate active matrix completion (MC)-based sampling for efficient and low cost MCS data gathering, we propose several novel techniques to solve this problem:

- After a thorough investigation of the relationship between sample locations and factor matrices in matrix factorization, we propose a ***bipartite graph*** to intelligently represent the environment matrix with well defined vertexes and edges. Moreover, to facilitate low cost sample selection and consider the incentives for MCS, we incorporate the sensing cost into the graph.
- Based on the well designed graph, we model the low cost sampling problem as an edge selection problem and view the matrix completion process as solving a linear system consisting of multiple equations.
- We propose ***two sample selection strategies*** to determine sample locations for accurate and robust matrix completion in the presence of unavoidable noise thus the perturbation to the sensory data. The first strategy does not schedule sampling at a position if this does not add much information for data recovery, while the second strategy provides a low-cost and efficient sample selection strategy to build a stable linear system by well utilizing the geometry features of the linear system.
- We provide a theoretical analysis to demonstrate that the linear systems in our solution are not under-determined and solvable, and the samples we take are sufficient to obtain the complete data even though the information of the upcoming time slot is unknown.
- We have performed extensive experiments using two data sets, one on PM 2.5 air conditions and the other on traffic speeds on the road. Our results demonstrate that our pre-scheduled scheme can recover the missing data at very high accuracy with the sampling ratio only around 11%. The peer matrix completion algorithms with similar recovery performance requires up to 4-9 times the number of samples of ours for the two public data sets.

The rest of the paper is organized as follows. We introduce the related work in Section 2. We present the problem in Section 3. We represent matrix factorization with bipartite graph in Section 4, and propose algorithms for sampling scheduling and graph-based matrix completion in Section 5 and Section 6, respectively. We give the algorithm analysis in Section 7. Finally, we evaluate the performance of the proposed algorithms through extensive experiments in Section 8, and conclude the work in Section 9.

2 RELATED WORK

One of the challenges faced by crowd sensing systems is the high burden on participants. The need of performing time and energy consuming data collection creates a barrier that prevents a large number of users from joining the systems. Financial incentives are often given for the participation [30, 49]. Although several studies [2, 11, 32, 47, 50] have proposed methods to minimize the number of redundant sensing tasks allocated or participants selected, an organizer has to collect at least one sample from each grid or get data from most of the cells in the target area [46, 47], which would incur a large sensing and communication cost. This will prevent the practical and large-scale deployment of MCS systems.

The progress of sparse techniques, such as compressive sensing [4] and matrix completion [5, 16, 21, 22, 42], facilitates several realizations of sparse MCS and sparse data gathering in wireless sensor networks [18, 26, 36, 43]. To infer the un-sampled data, these schemes usually work offline and exploit compressive sensing or matrix completion to provide the best estimation of the whole sensory data with a set of known entries in the data structure. However,

if the observed samples do not carry enough information, the recovery performance is very low which results in low data quality for MCS and WSN applications.

To make the samples sufficient for data recovery in real time, some recent studies are made to infer the missing data and obtain the complete data matrix on-line [38, 39, 44, 45]. In these schemes, samples are taken in multiple steps with a well designed sampling stop condition to reduce the total number of samples. More specifically, partial knowledge of current data can be obtained from previous steps, and additional samples are taken in the next step based on these data. However, the partial knowledge learning process usually causes a high computation cost. For example, to learn the partial knowledge, CCS-TA [37] and its extension SPACE-TA [39] design a leave-one-out re-sampling principle. Requiring many rounds of matrix completion operations, the computation cost is very high. Moreover, data values may vary in different steps. In order to integrate data sensed in several steps to form a data column during the reconstruction of matrix, such solutions face the time synchronization problem, which further reduces the data recovery quality.

Different from conventional matrix completion algorithms, we view the matrix completion process as solving a linear system consisting of multiple equations with each associated with one sample. We can identify the sufficient number of samples thus the number of equations to make the linear system not under-determined and solvable. Without a complex stop condition, our proposed AS-MCS can run online.

Instead of taking random samples, very few studies [6, 14, 31, 33, 34] investigate active sampling strategies to select sample locations. These methods first estimate the uncertainty or error of the missing entries in the matrix using the pre-recovered data, then actively select sample locations that have large estimated uncertainty or error. CCS-TA [37], SPACE-TA [39] and [38] have applied this method to identify the sample locations. However, this kind of sampling strategy has two main drawbacks:

First, the process of uncertainly calculation is separated from the process of matrix completion, thus the gain on matrix recovery with the selection of sampling locations can be hardly guaranteed. For example, although authors in [6] provide three methods (including conditional gaussian distribution, query by committee, and committee stability) to calculate the uncertainty of the missing data, we can not find the direct relationship between the uncertainty estimated and the quality of matrix completion.

Based on the linear system model of matrix completion, we propose two sample selection strategies in AS-MCS to select samples that can carry more information to the linear system for a higher gain in the recovery performance. We also provide theoretical analysis to demonstrate the effectiveness of such strategies even though the information of the upcoming time slot is unknown.

Second, the uncertainty calculation on the missing entries relies on pre-recovery of the missing entries. When the pre-recovery is not accurate, the uncertainty calculated would also suffer from a large error.

Rather than considering entries of an existing matrix as in [6, 38, 39], AS-MCS is designed to select future sample locations in a time slot which corresponds to a new column without any data yet. A typical matrix completion algorithm can not recover a matrix with a row or column completely empty, so existing active sampling algorithms based on the estimation of the uncertainty of un-sampled data is not applicable to schedule sampling in the future time.

In the cold start problem of a recommendation system, given a new user, her preferences on all items are inferred. This has some similarity to our MCS scenario if we consider each grid as an item and each time-slot as a user. For a new user, the system does not know any of her preferences. As a typical way [19, 23] to find user preference for all items, a new user is first queried for her interests for selected items. This active probing of information from human users does not apply to the MCS system we consider. Besides the time delay and bandwidth consumption, we cannot expect to overload users with constant responses. These user-dependent selection also does not help for improving the quality of matrix completion. Besides the query based algorithm, the active learning of sampling locations in [15] highly depends on the distribution of recommendation data. It can not be applied in the MCS monitoring whose data distribution is very different from that of the recommendation system. The sample selection in AS-MCS does not rely on data probing and pre-known data distribution.

Moreover, current sparsity-based data gathering schemes generally assume that the costs for collecting different data samples are always the same, regardless of where and who contribute the data. In practice, the data from different grids may not cost the same. For example, the sensing cost in a cell may be inversely proportional to the number of participants in the cell [25], or the network signal strength of the cell [48]. To provide incentives in MCS [13], unbalanced rewards are made to accommodate the cost difference in different geographic areas which are impacted by the economic feasibility, area coverage, and density of participants. To reduce the total sensing cost, the selection of sampling location needs to take the diverse grid costs into consideration. Without considering the cost in the sampling selection process, existing sparse MCS schemes may introduce high sensing cost.

Different from above studies, we propose an on-line scheme in AS-MCS to actively schedule the sampling process to ensure the high data recovery quality while taking into account diverse costs and noise in data samples. Rather than only taking random measurements following the basic MC theory, to further reduce the cost of MCS while ensuring the quality of missing data inference, we build a bipartite-graph to model the environment data and incorporate the diverse sensing cost into the graph. By associating the cost with the reward to the mobile users, our system naturally incorporates the incentives to encourage more device owners to participate in MCS and contribute to the efficient monitoring of a large system. Based on the graph, we further propose a bipartite-graph-based sensing scheduling scheme to actively determine the sampling positions in each upcoming time slot and a matrix completion algorithm to robustly and accurately recover un-sampled data.

3 PROBLEM

A major factor that prevents the large scale deployment of MCS is its sensing and communication cost, which discourages mobile users from participating in the sensing tasks. To facilitate the management of a MCS system, we divide a monitoring area into N grids, and record data collected over T time slots in an $N \times T$ matrix X. Data from a grid correspond to a row in the matrix, and a column of the matrix records data from all grids in a time slot.

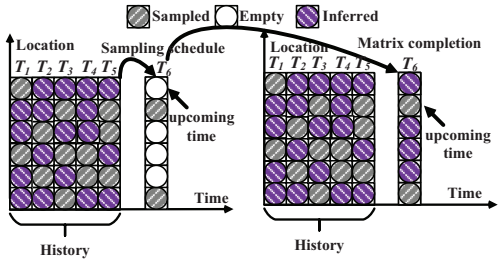


Figure 1: Sampling schedule problem

Our goal is to actively select the set of grids to monitor to minimize the sensing cost while also ensuring accurate inference of data in un-sampled grid locations. In the example of Fig. 1, we have historical measurement data in time slots T_1, T_2, T_3, T_4, and T_5, where the gray entries are the observed data and the un-sampled entries in purple are inferred based on these historical samples through matrix completion. We would like to design a sampling scheduling algorithm to determine where the measurements should be taken in the upcoming time slot T_6 such that its data in the remaining locations can be accurately recovered through matrix completion.

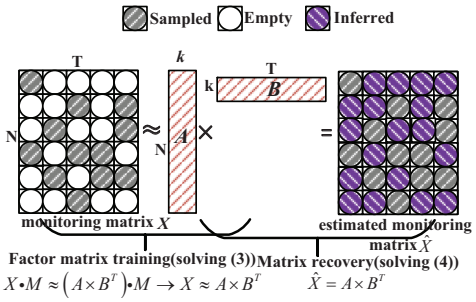


Figure 2: Matrix factorization model

On a typical MCS platform, mobile users that are willing to provide sensing data can register as candidate participants. To facilitate scheduling, participants can inform their grid locations to the MCS center, based on which MCS center can assign the sensing jobs to the selective participants based on their preferences and constraints. The mobility of users will be taken advantage to increase the sensing coverage, but we don't require users to move to specific positions with the scheduling. At any specific time, the scheduling is based on the current locations of available participants and the need for reliable inference of remaining data. As an inherent benefit of matrix completion, the data in the grids without participants can be inferred.

As an incentive, a participant may be given a reward for providing sensing data, financially or with credits. From each selected grid, MCS center can select one or more participants trading off between the sensing reliability and cost. If privacy is a concern, MCS center only needs to know the encoded IDs of participants not their physical identity. The privacy is not our focus. To provide the incentive, if an MCS center is informed with the minimum reward expected by each participant, it can assign the sensing tasks to minimize its total expense. Alternatively, the MCS center may announce the tasks with payment, and participants can compete in providing the services.

We use a Binary sampling matrix M to indicate the locations that are measured in the time slot T, where

$$m_{ij} = \begin{cases} 1 & \text{if location i at time j is sampled} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Let matrix P denote the data matrix that records the environment data of N grid locations over T time slots. Accordingly, the incomplete monitoring matrix X can be represented as

$$X_{N\times T} = P_{N\times T} \bullet M_{N\times T} \quad (2)$$

where $\bullet$ represents a scalar product (or dot product) of two matrices, with $x_{ij} = p_{ij}m_{ij}$.

Among various methods to infer the missing matrix items, we take the one based on matrix factorization. By factorizing a low-rank matrix into low-rank factor matrices, matrix factorization can capture the low rank feature of the matrix thus the spatio-temporal correlation for missing data recovery. We use Fig. 2 to illustrate the matrix factorization method. Given measurement samples, a monitoring matrix X with rank k is factored into the product of an $N \times k$ factor matrix A and a $T \times k$ factor matrix B by minimizing the loss as follows:

$$(A, B) = \arg\min_{A,B} \left\| \left(X - A \times B^T \right) \bullet M \right\|_F^2 \tag{3}$$

where the loss from matrix recovery is defined based on the Frobenius norm $\|\|_F$.

Given sufficient samples, many efforts (such as ALS [12] and SGD [24]) have been made to train the factor matrixes to solve the problem in (3). After obtaining the factor matrices A and B, the monitoring matrix can be estimated through

$$\hat{X} = A \times B^T \tag{4}$$

where $\hat{X}$ denotes the recovered monitoring matrix.

Existing methods (ALS and SGD) can achieve good recovery performance when the samples are sufficient, but will suffer from poor performance otherwise. From [6, 34], we know that different sampling locations bring different information to data recovery. Existing matrix completion algorithms only consider how to complete a matrix instead of determining where to take samples.

In contrast, we are interested in efficiently selecting locations to take samples in each upcoming time slot for low cost online MCS monitoring. Specifically, given that the sensing cost is a critical issue in MCS, we would like to investigate the possibility and methodology of making sampling schedule to minimize the sensing cost while accurately inferring the un-sampled data in presence of many practical constraints.

Let m_{T+1} denote the binary sampling column corresponding to the upcoming time slot $T+1$. Its i-th entry $m_{i,T+1}$ denotes whether the location i at time slot $T+1$ is sampled, and $w(i, T+1)$ denotes the cost to take this measurement. The total sensing cost in the upcoming time slot can be expressed as

$$\sum_{i=1}^{N} m_{i,T+1} w(i, T+1). \tag{5}$$

The key problem we want to solve is to identify the optimal m_{T+1} for each upcoming time slot so as to simultaneously achieve the following two goals:

(1) minimizing the sensing cost (i.e., Eq.(5))

(2) accurately inferring un-sampled data, that is, samples identified by locations $[M, m_{T+1}]$ can be applied to robustly solve the following matrix completion problem

$$(A, B) = \arg\min_{A,B} \left\| \left([X, x_{T+1}] - A \times B^T \right) \bullet [M, m_{T+1}] \right\|_F^2 \tag{6}$$

where x_{T+1} is the monitoring data at the time slot $T+1$, $A \in R^{N \times k}$ and $B \in R^{(T+1) \times k}$ are the factor matrices for the monitoring matrix $[X, x_{T+1}]$.

4 REPRESENTING MATRIX FACTORIZATION WITH BIPARTITE GRAPH

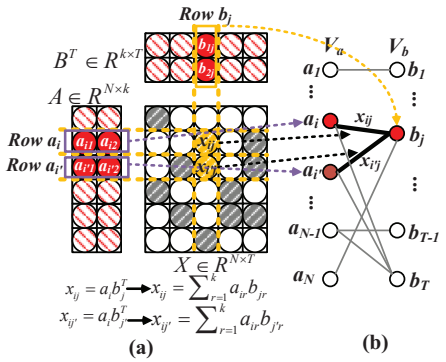


Figure 3: Matrix factorization and bipartite graph.

In order to schedule the future sampling, we would like to understand the relationship between each sample in the monitoring matrix and the entries in the factor matrices. From the definition of matrix factorization (Fig.2 and Fig.3(a)), each sample entry x_{ij} in the monitoring matrix can be written as $x_{ij} = a_i b_j^T$ where $a_i = \{a_{i1}, \cdots, a_{ik},\}$ and $b_j = \{b_{j1}, \cdots, b_{jk},\}$ are the i-th row and the j-th row of the factor matrices A and B, respectively. Easily, we can see that x_{ij} is a linear combination of entries in the factor matrices, where

$$x_{ij} = \sum_{r=1}^{k} a_{ir} b_{jr}, \tag{7}$$

where a_{ir}, b_{jr} are the r-th entry of the i-th row and j-th row of the factor matrices A and B, k is the rank of the monitoring matrix.

Taking $k = 2$ as an example, as shown in Fig.3, a sample (i, j) can be expressed as

$$x_{ij} = \sum_{r=1}^{k=2} a_{ir} b_{jr} = a_{i1} b_{j1} + a_{i2} b_{j2}. \tag{8}$$

Consequently, we can build a bipartite graph $G = \{V_a, V_b, E\}$ to represent the monitoring matrix. V_a, V_b are the vertex sets with each vertex denoting one row in the factor matrices A and B, respectively. If an entry x_{ij} is sampled with a measurement, an edge is added into the set E of the graph with

the vertexes a_i in V_a and b_j in V_b connected. One sample in the monitoring matrix corresponds to one edge in the graph. We use E to represent the edge set. As each sample is associated with one vertex in V_a and one vertex in V_b, there are no edges to connect vertexes within V_a or V_b.

Fig.3 shows the relationship between matrix factorization and bipartite graph. As factor matrices A and B have N rows and T rows respectively, in the bipartite graph, V_a and V_b respectively have N and T vertexes. A sample x_{ij} corresponds to an edge that connects a_i in V_a and b_j in V_b. As the first row in the monitoring matrix has one sample x_{11}, the graph has an edge connecting a_1 in V_a and b_1 in V_b.

According to the graph model, given a monitoring matrix X (rank $k = 2$) and its two samples (i, j) and (i', j), we can build a linear system with the following two equations:

$$x_{ij} = a_{i1}b_{j1} + a_{i2}b_{j2} \tag{9}$$

$$x_{i'j} = a_{i'1}b_{j1} + a_{i'2}b_{j2}, \tag{10}$$

where each sample corresponds to one equation. The linear system can be further written as follows

$$\begin{bmatrix} x_{ij} \\ x_{i'j} \end{bmatrix} = \begin{bmatrix} a_{i1} & a_{i2} \\ a_{i'1} & a_{i'2} \end{bmatrix} \begin{bmatrix} b_{j1} \\ b_{j2} \end{bmatrix} \tag{11}$$

If $\begin{bmatrix} x_{ij} \\ x_{i'j} \end{bmatrix}$ and $\begin{bmatrix} a_{i1} & a_{i2} \\ a_{i'1} & a_{i'2} \end{bmatrix}$ are known, as $\begin{bmatrix} b_{j1} \\ b_{j2} \end{bmatrix}$ has two unknown components (variables) b_{j1} and b_{j2}, it can be solved using two equations. Obviously, in (11), $\begin{bmatrix} b_{j1} \\ b_{j2} \end{bmatrix}$ corresponds to one row in the factor matrix B and a vertex in V_b.

The key problem of this work is to efficiently select samples and then train the two factor matrices using known samples to recover the missing data. From (11), we have the following observations:

- Training factor matrices with matrix completion process can be viewed as solving a linear system consisting of multiple equations corresponding to observed samples.
- For a matrix with the rank k, to train and determine an unknown row of a factor matrix with k unknown components, at least k equations are needed.
- The unknown row of a factor matrix corresponds to an unknown vertex in the graph. An equation in a linear system corresponds to a sample and thus an edge in the graph. Therefore, to make the linear system solvable, we require that the unknown vertex is connected with at least k known vertexes through k edges.

Based on above insights, we propose our sampling scheduling algorithm and matrix completion algorithm in Section 5 and Section 6, respectively.

According to (4), after factor matrices A and B are obtained, the monitoring matrix can be recovered. Let (i', j') denote a missing entry, the missing data can be estimated as

$$\hat{x}_{i'j'} = a_{i'}b_{j'}^T = \sum_{r=1}^{k} a_{i'r}b_{j'r}, \tag{12}$$

where $a_{i'}$ and $b_{j'}$ are the i'-th row and j'-th row in matrices A and B, respectively.

5 SAMPLING SCHEDULING BASED ON GRAPH

The goal of our sampling scheduling problem is to select the sampling locations for an upcoming time slot so that the complete sensing map can be inferred at high quality while minimizing the total cost.

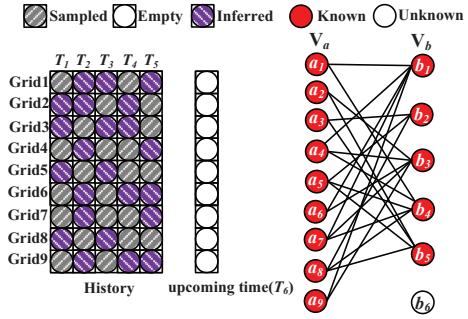


Figure 4: Graph corresponding to the sensing matrix.

Fig.4 gives an example to illustrate the relationship between historical data and the data in upcoming time slot, where the historical data are contained by a 9×5 matrix. Besides, we also use a column to represent the data of the upcoming time slot (T_6). As we don't know the data in the upcoming time slot in advance, the column is empty. When there are sufficient historical samples, all vertexes corresponding to the historical data in the bipartite can be trained and known. Therefore, in the bipartite graph, all vertexes in $V_a(a_1, \cdots, a_9)$ and the vertexes in $(b_1, b_2, \cdots, b_5)$ corresponding to historical data are marked "red" as known. Only the vertex b_{T+1} corresponding to the upcoming time slot $T + 1$ is unknown. Based on the graph, our sampling scheduling problem is transformed to selecting samples in the upcoming time slot (T_6) to be able to obtain b_{T+1}.

The degree of a vertex is the number of edges connected, and one sample corresponds to an edge. In Fig.4, as grid 1 has been monitored in time slots T_1 and T_4 with two samples $(1, 1)$ and $(1, 4)$, the graph has one edge connecting (a_1, b_1) and another connecting (a_1, b_4), respectively.

We associate each edge with a weight $w(i, j)$ to reflect the cost on monitoring the grid i at the time slot j. For grid locations that are not covered by candidate participants, we set the weights of the corresponding edges to large values to prevent assigning the monitoring tasks to the grids.

The sensing at different locations may involve a cost, either a physical cost associated with the sensing and communication or an economic cost to cover physical cost through a reward. How to estimate the sensing cost is not the focus of this paper. In our performance study in Section 8, following [25], the sensing cost in a cell is set inversely proportional to the number of participants in the cell. Such geographically unbalanced prices are reasonable from the perspective of economic market [49]. As pointed out in [49], due to the competitions among sellers (mobile users around), the benefit consumer MCS platform (buyer) can obtain is larger, as the average price asked by mobile users reduces when there are more users thus the potential sellers around.

According to the relationship between samples and the linear system, we can easily find that adding one sample means adding an equation to the linear system. For example, if we select grids 1, 2 to be sampled at time slot $T+1$, we can set up two equations corresponding to the two samples $(1, T+1)$ and $(2, T+1)$:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \end{bmatrix} \begin{bmatrix} b_{T+1,1} \\ b_{T+1,2} \\ \vdots \\ b_{T+1,k} \end{bmatrix} = \begin{bmatrix} x_{1,T+1} \\ x_{2,T+1} \end{bmatrix} \quad (13)$$

The problem of scheduling the sampling is transformed to the problem of adding k edges at the minimum cost to connect the unknown vertexes to the known ones, which will form a linear system to find the unknown vertexes.

To find the unknown vertex b_{T+1} with the linear system with the minimum cost, we can sort the vertex a_i in V_a according to $w(i, T+1)$ in an increasing order, and then select the first k edges to connect k vertexes in V_a to the vertex b_{T+1}. However, such a straightforward strategy may suffer from two problems that may make the linear system either *incomplete* or *unstable*, which fails to provide unique or stable solution. In following two subsections, we propose two sampling strategies to avoid these two problems.

5.1 Sampling to form a complete linear system

To determine an unknown vertex ($b \in \mathbb{R}^k$) with k unknown components, it requires at least k equations thus k samples to build the linear system $Ab = x$ to satisfy $rank\,[A] = k$. If the $rank\,[A] < k$, we call the linear system incomplete as it can not provide an unique solution.

To minimize the number of samples selected while avoiding the construction of an incomplete linear system, we design our ***sampling selection strategy 1*** described as follows. We denote the index set of the selected samples as Ω. To check if a new sampling point $x_{i,j}$ can be added to the set Ω, we will test whether its addition can increase the rank of the corresponding coefficient matrix A of the linear system built so far. If yes, $x_{i,j}$ can be added as a feasible sample; otherwise, we should search for another candidate sample.

When selecting the sampling locations, we don't know the sampling values but know part of the factor matrices corresponding to historical data. Fortunately, according to ***sampling selection strategy 1***, sampling location checking and selection involve only the coefficient matrix A. The coefficient matrix is built with the values of vertexes in V_a, which are known from the historical sampling data. Therefore, the above feasibility check is easily implemented.

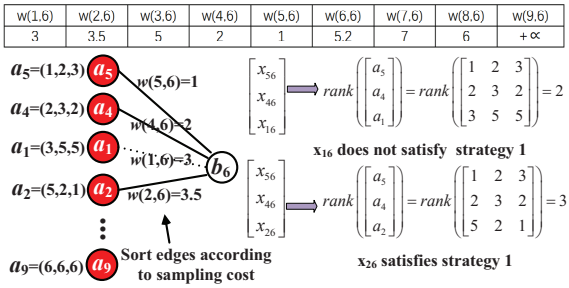


Figure 5: Sample selection for a complete system.

In Fig.5, x_{56} and x_{46} with the least sampling cost $w(4,6)$ and $w(4,6)$ are the samples selected in Ω. The next smallest cost for sampling is x_{16}. However, adding this sample can not increase the rank of the coefficient matrix and it is not a feasible sample. Instead, another point $x_{2,6}$ can be selected to increase the rank of the coefficient matrix.

5.2 Sampling to ensure the linear system to be robust

In MCS, due to the calibration errors of sensors or the unreliable transmission of data over wireless links, the sample data x may suffer from noise and even loss. A linear system $Ab = x$ is unstable if there exists a small perturbation ε in x, and the result from a linear system $Ab = x + \varepsilon$ could be significantly different. Therefore, when selecting the sampling points, it is critical to construct a robust linear system for the accurate inference of un-sampled measurement data.

We illustrate the un-stability problem in linear systems with an example. For a linear system $Ab = x$ with $\begin{bmatrix} 1 & 1 \\ 1.0001 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.0001 \end{bmatrix}$, we can easily see that the solution is $b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. If we use the slightly different right hand side vector $\hat{x} = \begin{bmatrix} 2.0001 \\ 2 \end{bmatrix}$ to form the linear system $A\hat{b} = \hat{x}$, we have the solution $\hat{b} = \begin{bmatrix} -1 \\ 3.0001 \end{bmatrix}$.

The linear systems $Ab = x$ and $A\hat{b} = \hat{x}$ have the same coefficient matrix A, but the small difference in the righthand-side vectors $\hat{x} - x = \begin{bmatrix} 0.001 \\ -0.001 \end{bmatrix}$ leads to completely different solutions b and $\hat{b}$. Following we analyze why the solution difference $b - \hat{b}$ happen in linear systems.

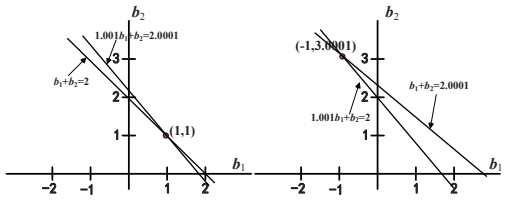


Figure 6: Example unstable linear system. (a) System I represents the lines $b_2 = 2 - b_1$ and $b_2 = 2.0001 - 1.0001b_1$, which intersect at $(1, 1)$ (b) System II represents the lines $b_2 = 2.0001 - b_1$ and $b_2 = 2 - 1.0001x_1$, which intersect at $(-1, 3.0001)$. The difference between the lines is amplified for illustration purpose.

Let b denote the solution vector of the linear system $Ab = x$. If we choose a slightly different right hand side vector $\hat{x}$, then we obtain a different vector $\hat{b}$ satisfying $A\hat{b} = \hat{x}$. We want to know how the relative error $\|\hat{x} - x\| / \|x\|$ influences the relative error $\left\|\hat{b} - b\right\| / \|b\|$ ("error propagation"). We have $A\left(\hat{b} - b\right) = (\hat{x} - x)$ and therefore,

$$\left\|\hat{b} - b\right\| = \left\|A^{-1}(\hat{x} - x)\right\| \leq \left\|A^{-1}\right\| \|(\hat{x} - x)\| \tag{14}$$

On the other hand, we have $\|x\| = \|Ab\| \leq \|A\| \|b\|$. Combining these two equations, we have $\frac{\left\|\hat{b}-b\right\|}{\|b\|} \leq \|A\| \left\|A^{-1}\right\| \frac{\|\hat{x}-x\|}{\|x\|}$.

The number $cond(A) = \|A\| \left\|A^{-1}\right\|$ is called condition number of matrix A. To ensure the solutions to be reliable, we need the condition number $cond(A) = \|A\| \left\|A^{-1}\right\|$ to be small. However, calculating the inverse of the coefficient matrix A involves a high computation cost. To provide a practical guidance for the sample point selection, we would like to resort to a geometric view of the linear system. In Fig.6, if two lines are almost parallel, a small change in the sampling value x can result in a relatively large distance between two intersection points from the two different linear systems.

Therefore, to make the solutions more reliable, we design our ***sampling selection strategy 2*** described as follows. To determine whether a new sampling point $x_{i,j}$ can be added to the set Ω, we will calculate the similarity between cosine values of the new candidate sampling point with all the points already selected, and check if the maximum cosine value is smaller than a threshold th. If yes, $x_{i,j}$ can be added as a feasible sample into Ω; otherwise, we should search for another candidate sample. For two samples (i, j) and (i', j), we can obtain the following two equations:

$$x_{ij} = a_{i1}b_{j1} + a_{i2}b_{j2} + \cdots + a_{ik}b_{jk} \tag{15}$$

$$x_{i'j} = a_{i'1}b_{j1} + a_{i'2}b_{j2} + \cdots + a_{i'k}b_{jk} \tag{16}$$

where k is the rank of matrix X. We denote the equation vectors of these two equations as following:

$$a_i = (a_{i1}, a_{i2}, \cdots, a_{ik}) \tag{17}$$

$$a_{i'} = (a_{i'1}, a_{i'2}, \cdots, a_{i'k}) \tag{18}$$

The cosine similarity, $cos(\theta)$, is represented as $\cos(\theta) = \left|\frac{a_i a_{i'}}{\|a_i\|\|a_{i'}\|}\right|$.

5.3 Complete solution

Considering both the feasibility and stability in selecting sample locations, we design the complete solution as follows.

1) To minimize the sampling cost, we first sort the candidate samples (edges) according to their edge weights in an increasing order. For samples with the same edge weight, its sorting is done according to the degrees of their corresponding vertexes in V_a.

2) We check one-by-one if a sample will ensure the linear system to be complete and robust, until we find k samples according to both ***sampling selection strategies 1 and 2***.

In step 1, when having the same edge weight, we sort samples according to the degrees of their corresponding nodes in V_a. This is because that the lower the degree of these vertexes in V_a, the fewer the historical samples have been used to train the corresponding rows in the factor matrix A. Therefore, adding more samples to associate with these low-degree vertexes can refine the corresponding parts in the factor matrix to capture more information for more accurate data recovery.

6 MATRIX COMPLETION BASED ON GRAPH

The coordination of sensing in MCS can be performed by a scheduler. A large geographic domain can be divided into smaller sub-domains, with a scheduler in each to quickly collect data. After selecting the set of sample locations for the upcoming time slot, the sensing tasks will be assigned to selected participants, and the matrix completion algorithm will be applied to infer un-sampled data after receiving the sensing data from these participants.

In each new time slot, the topology of the bipartite graph should be updated by adding a new vertex in V_b and new edges corresponding to new samples. As shown in Fig.7(a), vertex b_6 and edges connecting (a_6, b_6) and (a_9, b_6) are added. The connection relationship among old vertexes corresponding to historical data does not change.

To accurately recover the un-sampled data in the new time slot, we design a matrix completion algorithm based

on the bipartite graph in Algorithm 1, as illustrated in Fig.7. When two new samples (6,6) and (9,6) are gathered, the vertexes in V_a and V_b except b_6 are marked as known (shown in Fig.7(a)) from the training of historical data. According to step 1 in Algorithm 1, we can first build a linear system with these two new samples $(6, 6)$ and $(9, 6)$ to calculate b_6. After finding b_6, all vertexes in the graph are known in Fig.7(b). However, the vertexes except b_6 only include the old information brought by historical data instead of the new samples.

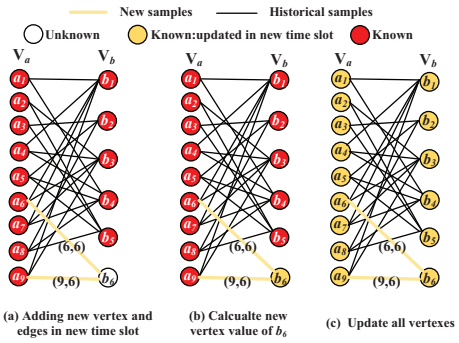


Figure 7: Processes in each new time slot

To address the issue, in every time slot, our matrix completion algorithm will update vertexes in V_a and V_b. As shown on lines 2-5 in Algorithm 1, to incorporate new information to all vertexes in V_a and V_b, we alternatively train vertexes in V_a and V_b using measurements taken in the new time slot together with the history samples until their values converge to stable ones. Finally, with all vertexes in the graph known, we can infer all the un-sampled data.

Algorithm 1 Matrix completion based on graph

1: Build the linear system using new samples to calculate the variables (corresponding to the unknown vertex in V_b)
2: **while** not convergent **do**
3: Fix the vertexes in V_b as known, treat vertexes in V_a as unknown variables, build the linear system using new samples together with historical samples to update the unknown variables
4: Fix the vertexes in V_a as known, treat vertexes in V_b as unknown variables, build the linear system using new samples together with historical samples to update the unknown variables
5: **end while**

With the update process on lines 2-5 in Algorithm 1, all vertexes' values in the graph are updated using the new measurement data. The update process is important for scheduling in future time slots, as the check of feasibility of selecting a sample only depends on the coefficient matrix built with vertexes in V_a. Incorporating new information at the current time slot to update vertexes in V_a can make future sample selection more accurate.

Obviously, the linear system for the update process is overdetermined with the number of equations larger than the number of variables as both historical data and new measurement data are counted. We denote such a linear system as $Dy = x$, where D is the coefficient matrix, y is the vector denoting variables unknown, x is a vector recording the measurement data. We propose a least square based algorithm, where the problem $Dy = x$ is expressed as $\min_y \|x - Dy\|$ to minimize the residual $x - Dy$. The solution of this can be written with the normal equations $y = \left(D^T D\right)^{-1} D^T x$ with T indicating a matrix transpose.

Although overcomplete linear systems may not be solvable using the conventional methods and usable for some applications, it brings two benefits to our problem: 1) New samples together with old samples can apply more information to better train the vertexes for more accurate data recovery; 2) It helps to alleviate the problems due to small noise and corruptions in the measurement data, which makes the data recovery more robust and accurate.

7 ALGORITHM ANALYSIS

In our sample scheduling algorithm, for each unknown vertex corresponding to an upcoming time slot, we will add samples to make the degree of the vertex reach k. Moreover, according to sample's feasibility check in Section 5.1, each sample added in our algorithm in Section5.3 should increase the rank of the coefficient matrix until k samples are selected and the matrix rank reaches k. Therefore, the linear system built to calculate the unknown vertexes on line 1 in Algorithm 1 is certainly not under-determined and is solvable. Moreover, as k unknown variables only needs k samples, the sample number under our scheme is the minimum.

Different locations selected for sampling build different linear systems in our algorithm, which further impacts the recovery performance. To obtain good recovery performance, **sample selection strategies 1 and 2** are adopted to select the locations to take measurements in MCS in the upcoming time slot. The sample locations that pass both the feasibility and stability check can be considered as the feasible ones.

In our scheduling algorithm, only the vertex information corresponding to sampling locations rather than the sensing value is needed. This good property makes our algorithm feasible in practical MCS systems, as the sensing values are unknown before taking measurements. In the following Theorem 1, we validate this property through theoretical analysis. Before we provide Theorem 1, we first present some definitions.

Definition 1 We denote by J the Jacobian of the mapping $\Upsilon : A, B \mapsto X = AB^T$. More specifically, the Jacobian of the mapping from A and B to X_{ij} can be written as follows:

$$\left(\frac{\partial x_{ij}}{\partial a_1}, \cdots, \frac{\partial x_{ij}}{\partial a_N}, \frac{\partial x_{ij}}{\partial b_1}, \cdots, \frac{\partial x_{ij}}{\partial b_T}\right) \tag{19}$$

where a_i is the i-th row vector of A and b_j is the j-th row vector of B. In (19), according to Eq(7), $\frac{\partial x_{ij}}{\partial a_{i'}} = \begin{cases} \mathbf{0} & i' \neq i \\ b_j & i' = i \end{cases}$ and $\frac{\partial x_{ij}}{\partial b_{j'}} = \begin{cases} \mathbf{0} & j' \neq j \\ a_i & j' = j \end{cases}$ where $\mathbf{0}$ denotes a vector whose size is k with each entry being 0. Therefore, Eq(19) can be written as

$$\begin{aligned} &\left(\frac{\partial x_{ij}}{\partial a_1}, \cdots, \frac{\partial x_{ij}}{\partial a_N}, \frac{\partial x_{ij}}{\partial b_1}, \cdots, \frac{\partial x_{ij}}{\partial b_T}\right) \\ = \quad & \underset{\substack{\uparrow \\ Derivative \quad wrt{:}a_i}}{(\mathbf{0} \cdots b_j \cdots \mathbf{0}} \qquad \underset{\substack{\uparrow \\ Derivative \quad wrt{:}b_j}}{\mathbf{0} \cdots a_i \cdots \mathbf{0})} \end{aligned} \tag{20}$$

Obviously, $\left(\frac{\partial x_{ij}}{\partial a_1}, \cdots, \frac{\partial x_{ij}}{\partial a_N}, \frac{\partial x_{ij}}{\partial b_1}, \cdots, \frac{\partial x_{ij}}{\partial b_T}\right)$ is a vector with the length equal to $k(N+T)$. As the matrix X has $N \times T$ entries and $X = AB^T$, we can write the Jacobian $J(A, B)$ as an $NT \times k(N+T)$ matrix.

$$J(A, B) = \begin{pmatrix} I_N \otimes b_1 & & \\ I_N \otimes b_2 & & \\ \vdots & & I_T \otimes A \\ I_N \otimes b_T & & \end{pmatrix} \tag{21}$$

where $\otimes$ denotes the Kronecker product. Here the rows of J correspond to the entries of X in the column major order.

Definition 2 For a position (i, j), we define $J_{(i,j)}$ to be the single row of J corresponding to the position (i, j). Similarly, we define J_E to be the submatrix of J consisting of rows corresponding to the set of observed positions E. J_E is the Jacobian of the mapping $\Omega \circ \Upsilon$ where Ω is the sampling location set.

The vector shown in Eq (20) can be denoted as $J_{(i,j)}$, which corresponds to the position (i, j). A matrix J_E formed with multiple vectors can be applied to represent the Jacobian of mapping for positions in E.

Let the set ε represent all the positions in a matrix X.

Definition 3 Let $E \subset \varepsilon$ be a set of observed positions. For the position $(i, j) \in \varepsilon \backslash E$, we say that the sample (i.e., x_{ij}) from this position is completable from samples taken at positions in E of a rank k matrix X if the entry x_{ij} is completable from $\Omega(X)$. The rank k finitely completable closure $cl_k(E)$ is the set of positions completable from E.

In Fig.8, the observed positions are contained in the set $E = \{(1,1), (1,4), (2,2), (2,4), (3,3), (4,2), (4,3)\}$. If we can infer values at positions $((1,2), (2,3), (4,4))$ based on these samples, then $cl_k(E) = \{(1,1), (1,4), (2,2), (2,4), (3,3), (4,2), (4,3), (1,2), (2,3), (4,4)\}$.

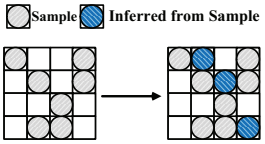


Figure 8: An example to illustrate $cl_k(E)$

Theorem 1 Let $E \in \varepsilon$ and let X be a rank k matrix. Then

$$cl_k(E) = \left\{(i, j) \in \varepsilon : J_{\{(i,j)\}} \in rowspan J_E\right\} \tag{22}$$

where $rowspan J_E$ is the set of all possible linear combinations of the row vectors in the Jacobian matrix J_E.

Proof Let $(i, j) \in \varepsilon \backslash E$. Before we provide the proof, we first utilize Fig.9 to illustrate some operations involved in the proof. Fig.9(a) shows the mapping $\Upsilon : A, B \mapsto X = AB^T$. $\Upsilon(A, B)$ is the resulted $N \times T$ matrix of the mapping. f is the projection of $\Upsilon(A, B)$ onto the set of entries at positions $E \cup \{(i, j)\}$ and g removes the entry (i, j). Fig.9(b) shows $f(\Upsilon(A, B))$. Fig.9(c) shows $g(f(\Upsilon(A, B)))$. g^{-1} is the reverse operation of g.

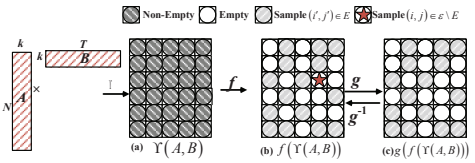


Figure 9: Operations

Let $(\mathbb{M}(N \times T, k))$ denote the space formed with matrices $\in \mathbb{R}^{N \times T}$ whose rank is not larger than k. According to the operations in Fig.9, we can find the open neighborhoods of $f(\Upsilon(A, B))$ (denoted as M) satisfying that $f(\Upsilon(A, B)) \in M \subset f(\mathbb{M}(N \times T, k))$. Similarly, we can also find the open neighborhoods of $g(f(\Upsilon(A, B)))$ (denoted as L) satisfying that $g(f(\Upsilon(A, B))) \in L \subset g \circ f(\mathbb{M}(N \times T, k))$ such that the restriction of g to M is smooth and $g^{-1}(L) \subset M$.

We have

$$\dim\left(g^{-1}(L)\right) + \dim N = \dim M \tag{23}$$

With the smoothness of $g(f(\Upsilon(A, B)))$ and $f(\Upsilon(A, B))$, according to constant rank theorem [27], we can obtain that:

$$\dim L = \dim\left(g(f(\Upsilon(A, B)))\right) = rank\left(J_E(A, B)\right) \tag{24}$$

and

$$\dim M = \dim\left(f(\Upsilon(A, B))\right) = rank\left(J_{E \cup \{(i,j)\}}(A, B)\right) \tag{25}$$

If and only if

$$rank\left(J_E(A, B)\right) = rank\left(J_{E \cup \{(i,j)\}}(A, B)\right), \tag{26}$$

we have $\dim\left(g^{-1}(L)\right) = 0$, that is, the position (i, j) is completable from E and $\Upsilon(A, B)$.

Equation (26) is just the assertion that $J_{\{(i,j)\}} \in rowspan J_E$. The proof completes.

Because the rows of J_E and $J_{E\cup\{(i,j)\}}$ have non-zero columns only at positions that depend on E and (i,j), whether (26) holds or not does not depend on the values of X but the sample positions. Therefore, without the knowledge of monitoring data value, we can schedule the sample selection procedure in advanced.

8 SIMULATION

We use one PM 2.5 air condition data set (denoted as PM 2.5) and one road traffic speed data set (denoted as Traffic) to evaluate our proposed AS-MCS.

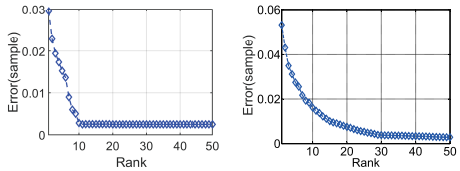
- PM 2.5 [53] includes PM 2.5 air condition data collected every one hour in the time span of 2014-05-01 to 2015-04-30 from 437 monitoring locations in 43 cities in China including Beijing, Tianjin, Guangzhou, Shenzhen, and other 39 adjacent cities.
- Traffic [1] includes traffic speed data collected from 142 road segments in Manhattan (New York City) every five minutes from 04:00 AM to 23:55 PM everyday during the time span from 2017-11-29 to 2018-01-11.

To evaluate the accuracy of different matrix completion algorithms, instead of absolute error metric, we use two relative error metrics $Error(sample) = \frac{\sqrt{\sum_{(i,j)\in\Omega}(x_{i,j}-\hat{x}_{i,j})^2}}{\sqrt{\sum_{(i,j)\in\Omega}(x_{i,j})^2}}$ and $Error(un-sample) = \frac{\sqrt{\sum_{(i,j)\in\overline{\Omega}}(x_{i,j}-\hat{x}_{i,j})^2}}{\sqrt{\sum_{(i,j)\in\overline{\Omega}}(x_{i,j})^2}}$ where $1 \le i \le N$, $1 \le j \le T$, Ω and $\overline{\Omega}$ denote the index sets of sample entries and un-sampled entries, respectively. $x_{i,j}$ and $\hat{x}_{i,j}$ denote respectively the (i,j)-th element of the raw environment matrix X and the matrix $\hat{X}$ recovered through the matrix completion. The first metric is the relative error to evaluate the impact of matrix completion on data elements whose initial values are observed, the second is the error for the inferred data.

8.1 Impact of matrix rank k

In our algorithm and other matrix completion algorithms based on matrix factorization, matrix rank is an important parameter. To investigate how matrix rank impacts the matrix factorization, given a rank, we first apply matrix factorization to the data sets PM 2.5 and Traffic to train the factor matrices and then recover the data (denoted by $\hat{X}$) with these trained factor matrices.

In Fig.10, we can see the error(sample) decreases with the increase of matrix rank, as an under-estimated matrix rank k makes the matrix factorization far from capturing the full structure of the data. After k reaches 10 (Traffic) and 30 (PM 2.5) respectively, increasing k will no longer increase the recovery performance. Therefore, setting $k = 10$ and $k = 30$ can capture the whole structure in the data. In the remaining simulations, we set $k = 10$ (Traffic) and $k = 30$ (PM2.5), respectively.



(a) Traffic (b) PM2.5

Figure 10: Impact of rank k.

In a practical application, the initial samples can be taken completely to estimate the matrix rank, and the rank can be adapted over time based on the relative recovery errors. The rank estimation is not the focus of this work. In our sample scheduling algorithm, for each unknown vertex corresponding to an upcoming time slot, we will add samples to make the degree of the vertex reach k. k directly determines the sampling cost. In practice, the sensory data may get loss due to the bad transmission condition. To prevent the problem, we can adopt a redundant sampling strategy. Therefore, in our following experiment, we will select $1.5 \times k$ samples for the upcoming time slot.

8.2 Effectiveness of AS-MCS

For each data set, the data of the first day are used for training the factor matrices to obtain the basic values of vertexes, and the data from the second day are used for testing. After the training, in each upcoming time slot, the following steps are taken: 1) using the trained vertexes to determine the sampling locations in the upcoming time slot, 2) running the matrix completion algorithm after collecting the data samples in the new time slot to infer the missing data in the time slot and update the vertexes in the graph, 3) going back to 1) using the updated vertexes to schedule the sensing in the next upcoming time slot.

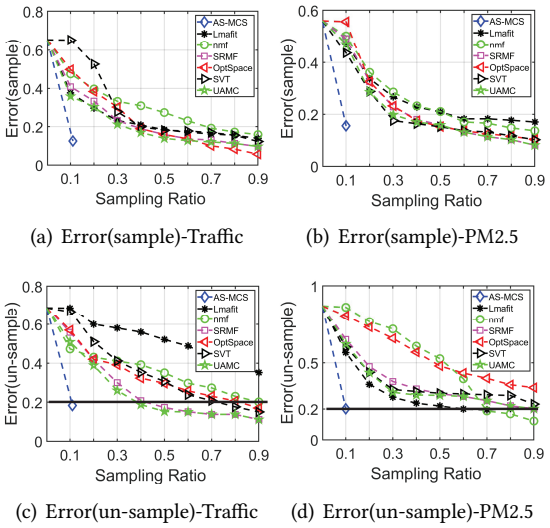
Besides our AS-MCS, we implement other five matrix completion algorithms *LMaFit*[40], *NMF*[20], *SRMF* [18, 51], *OptSpace*[17], and *SVT*[3] for the performance comparison. As these algorihtms do not support the sampling scheduling, for these matrix completion algorithms, we apply the uniform sampling to randomly choose the sample points in the 24 hours of the next day.

As discussed in Section 2, existing uncertainty based active sampling [6, 38] can only be applied to select samples within a matrix, and it can not be directly applied in our MCS scenario. To compare our AS-MCS with these existing algorithms, we adapt them so they can find samples for the upcoming time slot, and we call the revised algorithm

UAMC. More specifically, UAMC first uses the data in the last column of the matrix recovered from historical data as the pre-recovered data, and then applies the algorithm in [6, 38] to calculate the uncertainty and select samples to take in the upcoming time slot. Although authors in [6] provide three methods to calculate the uncertainty of the missing data, the method based on conditional gaussian distribution has been demonstrated in [6] to achieve the best recovery performance, so this method is taken in UAMC.

To evaluate how sampling costs from different locations impact the algorithms, following [25], we first randomly generate the number of participants following Poisson distribution in each grid for each time slot, then set the cost of each grid location as the reciprocal of the number of participants.

As all the matrix completion approaches are executed iteratively to train the parameters needed. For a fair comparison, we adopt the same two stop conditions: 1) The difference in the recovery loss on the sampling entries (i.e., $\frac{\sqrt{\sum_{(i,j)\in\Omega}(x_{i,j}-\hat{x}_{i,j})^2}}{N\times T}$) between two consecutive iterations is smaller than a given threshold value, set to 10^{-5} in this paper; 2) The maximum number of iterations is reached, set to 500 in this paper. The iteration process will continue until either of the two stop conditions is satisfied.



(a) Error(sample)-Traffic (b) Error(sample)-PM2.5

(c) Error(un-sample)-Traffic (d) Error(un-sample)-PM2.5

Figure 11: Recovery accuracy comparison.

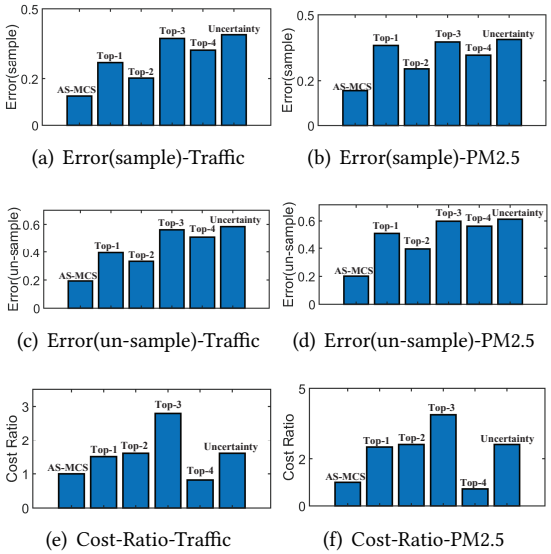
Fig.11 shows the recovery results for different data sets. The x-axis is the sampling ratio. Note that, for our AS-MCS scheme, we only show the upper bound on the sampling ratio thus sampling number. To investigate the recovery performance of other peer algorithms, we vary the sampling ratio.

From Fig.11, we observe that AS-MCS can achieve the low error ratio with a very small sampling ratio 11% for Traffic and 10% for PM2.5, while the errors of other algorithms come close to that of AS-MCS only when they use up to 4-9 times the number of samples at the sampling ratios 40%-90%. These results demonstrate the high effectiveness of AS-MCS in reducing the sampling rate and recovering data for different data sets.

The total sampling cost is obtained by summing up the sensing cost of all selected samples. To compare the sampling cost among different algorithms, given the total sampling cost under two algorithms (Alg_1 and Alg_2), we define the cost ratio as $\frac{C_{Alg_1}}{C_{Alg_2}}$ where C_{Alg_1} and C_{Alg_2} are the total sampling cost under Alg_1 and Alg_2. In this paper, we set our AS-MCS as Alg_2.

For a fair cost comparison, we use Table 1 to list the cost ratio and sampling ratio when all algorithms implemented achieve the relative error(un-sample) 0.2. If the peer algorithms can not reach the relative error even when the sampling ratio reaches 90%, we use – to fill the corresponding table entry. We find that AS-MCS can have significantly smaller sample ratio and sampling cost compared with other peer algorithms under all simulation scenarios. In Table 1, taking the results from the traffic dataset as an example, the costs under nmf, SRMF, OptSpace, SVT, UAMC are up to 69, 32.96, 63.63, 53.66, and 28.36 times that under our AS-MCS.

8.3 Effectiveness of the sampling points selected



(a) Error(sample)-Traffic (b) Error(sample)-PM2.5

(c) Error(un-sample)-Traffic (d) Error(un-sample)-PM2.5

(e) Cost-Ratio-Traffic (f) Cost-Ratio-PM2.5

Figure 12: Validate the effectiveness of the sampling points selected.

		AS-MCS	Lmafit	nmf	SRMF	OptSpace	SVT	UAMC
Traffic	Sample-Ratio	0.11	-	0.9	0.43	0.83	0.7	0.37
	Cost-Ratio	1	-	69	32.96	63.63	53.66	28.36
PM2.5	Sample-Ratio	0.1	0.57	0.7	0.9	-	-	0.9
	Cost-Ratio	1	60.16	73.88	95.01	-	-	95.01

Table 1: Cost-Ratio comparison.

To validate the effectiveness of our sampling selection algorithm, we implement other five sampling selection algorithms along with our matrix completion algorithm. We call the number of sample points selected in AS-MCS NUM. For a fair performance comparison, all sample selection algorithms select NUM samples. The peer sampling algorithms are described as follows. 1) Uncertainty: is an adaption of the uncertainty-based active sampling algorithm [6] (described in Section 8.2); 2) Top-1: selects the top-NUM positions with the fewest number of historical samples; 3) Top-2: selects the top-NUM positions whose historical samples have the highest variance; 4) Top-3: selects the top-NUM positions with the largest edge weights; 5) Top-4: selects the top-NUM positions with the smallest edge weights.

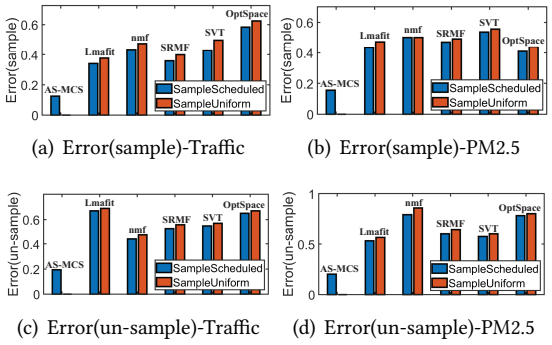
As shown in Fig.12, besides our sampling selection algorithm, all other sampling strategies (Top-1, Top-2, Top-3, Top-4 and Uncertainty) suffer from high recovery errors as they can not guarantee the building of a complete and stable linear system for robust and accurate matrix completion. Although Uncertainty is an active sampling algorithm, as it can not select samples that bring more information to our matrix completion algorithm, the recovery performance under Uncertainty is even worse than that under Top-1, Top-2, Top-3, Top-4.

Fig.12(e)(f) show the cost ratio in the simulations. Our AS-MCS achieves the lowest error ratio thus the best recovery performance with the sensing cost slightly larger than that under Top-4. Although Top-4 achieves the lowest sensing cost as it selects samples with the smallest cost, its recovery error is more than 2 times that of our AS-MCS.

8.4 Effectiveness of the proposed matrix completion algorithm

To validate the effectiveness of our proposed matrix completion algorithm, we conduct the following simulations. We feed the sample points selected in our AS-MCS to other matrix completion algorithms (denoted as SampleScheduled in Fig.13). For a performance comparison, we also randomly select NUM sample points following the uniform sampling, and feed these samples to other matrix completion algorithms (denoted as SampleUniform in Fig.13).

As shown in Fig.13, compared with other peer matrix completion algorithms, our graph-based matrix completion is very effective in inferring the missing data and can significantly increase the recovery accuracy with lower recovery errors. The recovery error of AS-MCS is less than 50% those of other methods. Moreover, the recovery error under SampleScheduled is smaller than that under SampleUniform in all cases, which demonstrates that the sampling selected in our algorithm can exploit the information hidden in the air and traffic monitoring data for more accurate missing data inference.



(a) Error(sample)-Traffic (b) Error(sample)-PM2.5

(c) Error(un-sample)-Traffic (d) Error(un-sample)-PM2.5

Figure 13: Validate the effectiveness of the proposed matrix completion algorithm.

9 CONCLUSION

In this paper, we propose an Active Sparse MCS scheme (AS-MCS) based on matrix completion to minimize the sensing cost while accurately inferring the un-sampled monitoring data. Three novel techniques are proposed in AS-MCS. 1) We propose a ***bipartite graph*** to intelligently represent the environment matrix. 2) We develop a sampling scheduling algorithm to carefully select the grid locations to take samples in each upcoming time slot, taking into account the difference in information, cost for the sensing of different grid locations and the impact of the samples on the reliability of solutions. 3) We propose a graph-based matrix completion algorithm to accurately and robustly recover the un-sampled data in the presence of data perturbation as a result of sensing and communication errors. Extensive performance studies using public data sets demonstrate that our AS-MCS scheme can recover the missing data at very high accuracy with the sampling ratio only around 11%.

ACKNOWLEDGMENT

The work is supported by the National Natural Science Foundation of China under Grant Nos.61572184 and 61725206, Hunan Provincial Natural Science Foundation of China under Grant No.2017JJ1010, U.S. NSF ECCS 1731238 and NSF CNS 1526843, the open project funding (CASNDST201704) of CAS Key Lab of Network Data Science and Technology, and the open project funding (CARCH201809) of State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences, Hunan Provincial Innovation Foundation For Postgraduate (CX2018B227), and the China Scholarship Council Foundation (201806130133).

REFERENCES

[1] [n. d.]. nyctmc. http://flowmap.nyctmc.org/weborb4/flowmap.

[2] Asaad Ahmed, Keiichi Yasumoto, Yukiko Yamauchi, and Minoru Ito. 2011. Distance and time based node selection for probabilistic coverage in people-centric sensing. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*. IEEE, 134–142.

[3] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization* 20, 4 (2010), 1956–1982.

[4] Emmanuel J Candès et al. 2006. Compressive sampling. In *Proceedings of the international congress of mathematicians*, Vol. 3. Madrid, Spain, 1433–1452.

[5] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9, 6 (2009), 717–772.

[6] Shayok Chakraborty, Jiayu Zhou, Vineeth Balasubramanian, Sethuraman Panchanathan, Ian Davidson, and Jieping Ye. 2013. Active matrix completion. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, 81–90.

[7] Yohan Chon, Nicholas D Lane, Yunjong Kim, Feng Zhao, and Hojung Cha. 2013. Understanding the coverage and scalability of place-centric crowdsensing. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 3–12.

[8] Haipeng Dai, Qiufang Ma, Xiaobing Wu, Guihai Chen, David KY Yau, Shaojie Tang, Xiang-Yang Li, and Chen Tian. 2018. CHASE: Charging and Scheduling Scheme for Stochastic Event Capture in Wireless Rechargeable Sensor Networks. *IEEE Transactions on Mobile Computing* (2018).

[9] Haipeng Dai, Xiaobing Wu, Guihai Chen, Lijie Xu, and Shan Lin. 2014. Minimizing the number of mobile chargers for large-scale wireless rechargeable sensor networks. *Computer Communications* 46 (2014), 54–65.

[10] Raghu K Ganti, Fan Ye, and Hui Lei. 2011. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* 49, 11 (2011), 32–39.

[11] Sara Hachem, Animesh Pathak, and Valérie Issarny. 2013. Probabilistic registration for large-scale mobile participatory sensing. In *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*. IEEE, 132–140.

[12] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 549–558.

[13] Luis G Jaimes, Idalides J Vergara-Laurens, and Andrew Raij. 2015. A survey of incentive techniques for mobile crowd sensing. *IEEE Internet of Things Journal* 2, 5 (2015), 370–380.

[14] Rong Jin and Luo Si. 2004. A Bayesian approach toward active learning for collaborative filtering. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 278–285.

[15] Rasoul Karimi, Christoph Freudenthaler, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2011. Non-myopic active learning for recommender systems based on matrix factorization. In *Information Reuse and Integration (IRI), 2011 IEEE International Conference on*. IEEE, 299–303.

[16] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. 2010. Matrix completion from a few entries. *IEEE Transactions on Information Theory* 56, 6 (2010), 2980–2998.

[17] Sewoong Keshavan, Raghunandan H.and Oh and Andrea Montanari. 2009. Matrix Completion from a Few Entries. *Information Theory IEEE Transactions on* 56, 6 (2009), 2980–2998.

[18] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Min-You Wu, and Xue Liu. 2013. Data loss and reconstruction in sensor networks. In *IEEE INFOCOM*.

[19] J Langford and T Zhang. 2008. The epoch-greedy algorithm for contextual multi-armed bandits. *Advances in Neural Information Processing Systems* 20 (2008), 817–824.

[20] Daniel D Lee and H Sebastian Seung. 2001. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*. 556–562.

[21] Hao Li, Kenli Li, Jiyao An, and Keqin Li. 2018. MSGD: a novel matrix factorization approach for large-scale collaborative filtering recommender systems on GPUs. *IEEE Transactions on Parallel and Distributed Systems* 29, 7 (2018), 1530–1544.

[22] Hao Li, Keqin Li, Jiyao An, Weihua Zheng, and Kenli Li. 2018. An efficient manifold regularized sparse non-negative matrix factorization model for large-scale recommender systems on GPUs. *Information Sciences* (2018).

[23] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.

[24] Yongjun Liao, Wei Du, Pierre Geurts, and Guy Leduc. 2013. DMFSGD: A decentralized matrix factorization algorithm for network distance prediction. *IEEE/ACM Transactions on Networking* 21, 5 (2013), 1511–1524.

[25] Yan Liu, Bin Guo, Yang Wang, Wenle Wu, Zhiwen Yu, and Daqing Zhang. 2016. Taskme: multi-task allocation in mobile crowd sensing. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 403–414.

[26] Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen. [n. d.]. Compressive data gathering for large-scale wireless sensor networks. In *ACM MOBICOM 2009*.

[27] Paul Malliavin. 1972. Géométrie différentielle intrinseque. (1972).

[28] Rajib Kumar Rana, Chun Tung Chou, Salil S Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-phone: an end-to-end participatory urban noise mapping system. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*. ACM, 105–116.

[29] Benjamin Recht. 2011. A simpler approach to matrix completion. *Journal of Machine Learning Research* 12, Dec (2011), 3413–3430.

[30] Sasank Reddy, Deborah Estrin, Mark Hansen, and Mani Srivastava. 2010. Examining micro-payments for participatory sensing data collections. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 33–36.

[31] Irina Rish and Gerald Tesauro. 2008. Active Collaborative Prediction with Maximum Margin Matrix Factorization. *ISAIM* 2008 (2008), 20.

[32] Xiang Sheng, Jian Tang, and Weiyi Zhang. 2012. Energy-efficient collaborative sensing with mobile phones. In *INFOCOM, 2012 Proceedings IEEE*. IEEE, 1916–1924.

[33] Jorge Silva and Lawrence Carin. 2012. Active learning for online bayesian matrix factorization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 325–333.

[34] Dougal J Sutherland, Barnabás Póczos, and Jeff Schneider. 2013. Active learning and search on low-rank matrices. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 212–220.

[35] Mehmet C Vuran, Özgür B Akan, and Ian F Akyildiz. 2004. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks* 45, 3 (2004), 245–259.

[36] Jin Wang, Shaojie Tang, Baocai Yin, and Xiang-Yang Li. [n. d.]. Data gathering in wireless sensor networks through intelligent compressive sensing. In *IEEE INFOCOM 2012*.

[37] Leye Wang, Daqing Zhang, Animesh Pathak, Chao Chen, Haoyi Xiong, Dingqi Yang, and Yasha Wang. 2015. CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 683–694.

[38] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah M'hamed. 2016. Sparse mobile crowdsensing: challenges and opportunities. *IEEE Communications Magazine* 54, 7 (2016), 161–167.

[39] Leye Wang, Daqing Zhang, Dingqi Yang, Animesh Pathak, Chao Chen, Xiao Han, Haoyi Xiong, and Yasha Wang. 2017. SPACE-TA: Cost-Effective Task Allocation Exploiting Intradata and Interdata Correlations in Sparse Crowdsensing. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 2 (2017), 20.

[40] Zaiwen Wen, Wotao Yin, and Yin Zhang. 2012. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation* 4, 4 (2012), 333–361.

[41] Kun Xie, Xueping Ning, Xin Wang, Shiming He, Zuoting Ning, Xiaoxiao Liu, Jigang Wen, and Zheng Qin. 2017. An efficient privacy-preserving compressive data gathering scheme in WSNs. *Information Sciences* 390 (2017), 82–94.

[42] Kun Xie, Xueping Ning, Xin Wang, Dongliang Xie, Jiannong Cao, Gaogang Xie, and Jigang Wen. 2017. Recover corrupted data in sensor networks: A matrix completion solution. *IEEE Transactions on Mobile Computing* 16, 5 (2017), 1434–1448.

[43] Kun Xie, Lele Wang, Xin Wang, Jigang Wen, and Gaogang Xie. 2014. Learning from the past: Intelligent on-line weather monitoring based on matrix completion. In *Distributed Computing Systems (ICDCS), 2014 IEEE 34th International Conference on*. IEEE, 176–185.

[44] Kun Xie, Lele Wang, Xin Wang, Gaogang Xie, and Jigang Wen. 2018. Low cost and high accuracy data gathering in WSNs with matrix completion. *IEEE Transactions on Mobile Computing* 17, 7 (2018), 1595–1608.

[45] Kun Xie, Lele Wang, Xin Wang, Gaogang Xie, Guangxing Zhang, Dongliang Xie, and Jigang Wen. 2015. Sequential and adaptive sampling for matrix completion in network monitoring systems. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2443–2451.

[46] Haoyi Xiong, Daqing Zhang, Guanling Chen, Leye Wang, Vincent Gauthier, and Laura E Barnes. 2016. iCrowd: Near-optimal task allocation for piggyback crowdsensing. *IEEE Transactions on Mobile Computing* 15, 8 (2016), 2010–2022.

[47] Haoyi Xiong, Daqing Zhang, Leye Wang, and Hakima Chaouchi. 2015. Emc 3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint. *IEEE Transactions on Mobile Computing* 14, 7 (2015), 1355–1368.

[48] Liwen Xu, Xiaohong Hao, Nicholas D Lane, Xin Liu, and Thomas Moscibroda. 2015. Cost-aware compressive sensing for networked sensing systems. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 130–141.

[49] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. 2012. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 173–184.

[50] Daqing Zhang, Haoyi Xiong, Leye Wang, and Guanling Chen. 2014. CrowdRecruiter: selecting participants for piggyback crowdsensing under probabilistic coverage constraint. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 703–714.

[51] Yin Zhang, Matthew Roughan, Walter Willinger, and Lili Qiu. 2009. Spatio-temporal compressive sensing and internet traffic matrices. In *In SIGCOMM'09:Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 267–278.

[52] Yu Zheng, Tong Liu, Yilun Wang, Yanmin Zhu, Yanchi Liu, and Eric Chang. 2014. Diagnosing New York city's noises with ubiquitous data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 715–725.

[53] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting Fine-Grained Air Quality Based on Big Data. In *SIGKDD (KDD '15)*. ACM, New York, NY, USA, 10. https://doi.org/10.1145/2783258.2788573