Low Cost and High Accuracy Data Gathering in WSNs with Matrix Completion

Kun Xie, Lele Wang, Xin Wang, *Member, IEEE*, Gaogang Xie, and Jigang Wen

Abstract—Matrix completion has emerged very recently and provides a new venue for low cost data gathering in Wireless Sensor Networks (WSNs). Existing schemes often assume that the data matrix has a known and fixed low-rank, which is unlikely to hold in a practical system for environment monitoring. Environmental data vary in temporal and spatial domains. By analyzing a large set of weather data collected from 196 sensors in ZhuZhou, China, we reveal that weather data have the features of low-rank, temporal stability, and relative rank stability. Taking advantage of these features, we propose an on-line data gathering scheme based on matrix completion theory, named MC-Weather, to adaptively sample different locations according to environmental and weather conditions. To better schedule sampling process while satisfying the required reconstruction accuracy, we propose several novel techniques, including three sample learning principles, an adaptive sampling algorithm based on matrix completion, and a uniform time slot and cross sample model. With these techniques, our MC-Weather scheme can collect the sensory data at required accuracy while largely reducing the cost for sensing, communication, and computation. We perform extensive simulations based on the data traces from weather monitoring and the simulation results validate the efficiency and efficacy of the proposed scheme.

Index Terms—Sparse data gathering, matrix completion, wireless sensor network

1 Introduction

Wireless sensor networks (WSNs) [1], [2], [3] are widely utilized to gather environmental information. Some example applications include the monitoring under the water, in the forest, and on the volcano. The data collected from the monitoring of the varying environment can generally be represented by an $N \times T$ Environment Matrix, which records data from N sensors over T time slots.

To obtain the environment matrix, in the traditional data gathering approach [4], a sensor node senses and sends data to a sink every time slot, which leads to a large amount of traffic and high sensing cost. Since the sensor nodes usually have limited computing ability and power supply, a primary goal of environment monitoring is to collect the sensory data at required accuracy with the least energy consumption.

To reduce the communication cost, some conventional methods have been proposed in WSN, such as distributed source coding techniques [5], [6], [7], in-network collaborative wavelet transform [8], [9], and data aggregation [10], [11], [12], [13], [14], [15], [16]. These methods exploit the spatial correlation in sensory data at sink or sensor nodes, but they may bring extra computational and communication overheads.

Recently, the compressive sensing (CS) theory provides a new paradigm for data gathering in WSNs [17], [18], [19], [20], [21], [22], [23], [24]. Although CS-based approaches can save energy and reduce sensing cost, they are originally designed to recover the sparse vector such as events. Some applications do not have clear sparsity features, and in many cases, we need to get more complete data which can be formed in the matrix style such as an Environment Matrix rather than just events for system management purpose.

With the rapid progress of sparse representation, matrix completion [25], [26], [27], a remarkable new field, has emerged very recently. According to the matrix completion theory, a low-rank matrix can be accurately reconstructed with a relatively small number of entries in the matrix. With matrix completion, only a small set of samples need to be taken by sensor nodes, which will not incur excessive computational and traffic overheads at resource limited sensor nodes in WSNs. Therefore, matrix completion provides a new venue for low cost data gathering.

In matrix completion, low-rank is necessary for accurate reconstruction of measured data and the rank of the matrix directly impacts the number of samples required to take. Existing matrix completion solutions often assume that the data matrix has a known and fixed low-rank, and therefore the number of measurements to take is fixed and determined by the relation between the smallest required number of samples and the rank of the matrix r. Unfortunately, such an assumption is unlikely to hold for data gathering in a practical and dynamic environment, and our observation

- *K. Xie is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410082, China, and the Key Laboratory of Machine Intelligence and Advanced Computing, (Sun Yat-sen University), Ministry of Education, Guangzhou 510000, China. E-mail: xiekun@hnu.edu.cn.*
- *L. Wang is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410082, China. E-mail: cswangle89@gmail.com.*
- *X. Wang is with the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794. E-mail: xwang@ece.sunysb.edu.*
- *G. Xie and J. Wen are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China. E-mail: {xie, wenjigang}@ict.ac.cn.*

Manuscript received 27 Nov. 2014; revised 23 June 2017; accepted 2 Nov. 2017. Date of publication 20 Nov. 2017; date of current version 1 June 2018. (Corresponding author: Kun Xie.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2017.2775230

on weather data trace indicates that the rank of the environment matrix varies over time.

To study the feature of environment data, we have deployed 196 sensors in Zhu Zhou, China, to collect weather data for more than two years. From the large weather data trace collected, we find that the rank of weather data may change with time, thus existing matrix completion solutions will not perform well. For example, if the rank of the data matrix increases, more measurements are needed for accurate reconstruction, and the reconstruction may fail otherwise. Therefore, to handle dynamic changes in weather data, it is desirable for the on-line data gathering system to adapt the number of samples to take.

In this paper, we first analyze large traces of real weather data, and our study reveals that there exist hidden structures in the data. By taking advantage of these structures, we propose an on-line data gathering scheme based on matrix completion theory, named MC-Weather, which can adaptively sample different locations in response to changes in environment and weather conditions. We propose several novel techniques to well schedule the sampling process while satisfying the required accuracy for matrix reconstruction. Because only a subset of locations are sampled, our MC-Weather scheme can largely reduce the amount of traffic and computation cost. Our contributions are summarized as follows:

- We have analyzed the traces of a large set of real weather data, and our results reveal that weather data have the features of low-rank, temporal stability, and relative rank stability. We also prove that the observed relative rank stability is a common feature in continuous data gathering systems.
- Taking advantage of the relative rank stability feature, we propose three sample learning principles, based on which we propose an adaptive sampling algorithm to quickly find an effective set of samples to take while the complete measurement data are recovered from matrix completion.
- To take the full advantage of our sample learning principle, we propose a Uniform Time-Slot and Cross Sample model (UTSCS). Compared with the Bernoulli model, we prove that our model ensures the matrix to have better features for higher matrix completion performance.
- Through comprehensive simulations with real data traces, we show that our MC-Weather scheme can accurately acquire weather data at very low cost, which significantly outperforms the competing methods.

To the best of our knowledge, this is the first work that proposes an adaptive matrix completion algorithm for low-cost on-line data gathering in dynamic environment. Our data gathering scheme is designed to be general without relying on specific matrix reconstruction algorithm or assuming the knowledge of the sparsity level of the unknown data.

We call our scheme MC-Weather because this paper utilizes weather data gathering as a case to verify the proposed data gathering scheme. Besides environment monitoring in WSNs, our scheme is flexible to apply in various networked monitoring systems including the monitoring of smart grid [28], [29], data center [30], social network [31], and other infrastructure.

The rest of this paper is organized as follows. We introduce the related work in Section 2. The fundamentals of matrix completion and problem formulation are presented in Section 3. We present our empirical study with real weather data in Section 4. The proposed MC-Weather is presented in Section 5. Finally, we evaluate the performance of the proposed MC-Weather through extensive simulations in Section 6, and conclude the work in Section 7.

2 Related Work

Structure and redundancy in data are often synonymous with sparsity. There exist two typical sparsity representation techniques, compressive sensing and matrix completion. In this section, we review related work and identify the differences of our work from existing work.

Compressive Sensing is a technique that can accurately recover a vector from a subset of samples given that the vector is sparse [18], [32], [33], [34] with only a few nonzero elements. The fundamental works of CS include the introduction of the l_1-minimization method to reconstruct the sparse vector. Later works on the reconstruction techniques provide some greedy approaches [35], [36], [37], [38] to recover the components of the vector gradually. Compressive sensing has two features, universal sampling and decentralized simple encoding, which makes it a new paradigm for data gathering in sensor networks [17], [18], [19], [23], [24]. Due to wireless transmission interference [39], [40], data loss is unavoidable in the process of data gathering. Moreover, as a powerful and generic technique for estimating missing data, CS has been applied to estimate the lost data [41].

The majority of work on CS consider vectors of data. A naive approach to deal with matrices might be to transform these matrices into vectors and then apply vector techniques. Compared with the vector-based recovery approaches, as a matrix could capture more information and the correlation among the sensor data in two dimension, matrix-based approaches can achieve much better recovery performance.

On the heels of compressed sensing, matrix completion has emerged very recently [25], [26], [27], [42], [43], [44]. Candès et al. [25] show that most $n_1 \times n_2$ matrices of rank r ($r \ll \min\{n_1, n_2\}$) can be perfectly recovered with very high probability by solving a simple convex optimization program provided that the number of samples is sufficient. New results show that matrix completion is provably accurate even when the few observed entries are corrupted with noises [42], [45]. Matrix completion brings new opportunities to fully exploit the low-rank property in various associated applications [43], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60].

Existing schemes based on matrix completion are mostly designed for off-line execution and can not be applied to on-line gathering of varying environment data. Moreover, existing algorithms determine the number of measurements assuming the rank of data matrix is known and does not change. This makes these algorithms difficult to apply to the practical monitoring of dynamic environment where the rank of the measurement matrix changes over time.

In this work, we propose an adaptive algorithm which can respond to the environment changes to intelligently determine the number of samples to take in a specific time

slot based on past monitoring data and matrix reconstruction accuracy requirement. We propose different strategies to facilitate the learning process for high quality and low cost environment monitoring.

3 Preliminary and Problem Formulation

In this section, we first introduce the fundamentals of matrix completion, then present our problem formulation.

3.1 Fundamentals of Matrix Completion

Matrix completion is a new technique which can be applied to recover a low-rank matrix from a subset of the matrix entries [25], [26], [27], [42]. That is, an unknown matrix $M \in R^{n_1 \times n_2}$ with rank $r \ll \min\{n_1, n_2\}$ can be recovered if a subset of its entries $M_{ij}, (i,j) \in \Omega$ are known and randomly selected from the matrix. The sampling operator $P_\Omega : R^{n_1 \times n_2} \rightarrow R^{n_1 \times n_2}$ is defined by

$$[P_\Omega(X)]_{ij} = \begin{cases} X_{ij} & (i,j) \in \Omega \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

If the set Ω contains enough information, there is a unique rank-r matrix that is consistent with the observed entries and can be recovered by solving the following rank minimization problem [25]

$$\begin{aligned} &\min\ rank(X) \\ &\text{subject to } P_\Omega(X) = P_\Omega(M), \end{aligned} \tag{2}$$

where the rank(.) denotes the rank of a matrix X.

However, solving this rank minimization problem in (2) is often impractical because it is NP-hard. Then [25] proves that most matrices M of rank r can be perfectly recovered by solving the optimization problem

$$\begin{aligned} &\min\ \|X\|_* \\ &\text{subject to } P_\Omega(X) = P_\Omega(M), \end{aligned} \tag{3}$$

provided that the number of samples m be sufficient and meet the following condition:

$$m \geq Cn^{6/5} r \log n, \tag{4}$$

where C is a numerical constant and $n = \max\{n_1, n_2\}$.

In (3), $\|X\|_*$ is the nuclear norm of the matrix X, which is the sum of its singular values. That is, $\|X\|_* = \sum_{i=1}^{\min\{n_1,n_2\}} \sigma_i$ and $\sigma_i \geqslant 0$ are the singular values of X.

Many approaches have been proposed to solve the convex optimization problem in (3), including iterative reweighted least squares algorithm (IRLS-M) [61], Spectral Matrix Completion[62], fixed point continuation algorithm [63], OptSpace [62], FixedPoint Continuation with Approximate SVD (FPCA) [64], and singular value thresholding (SVT) [65]. These algorithms use the observed entries as the training data to derive the parameters needed, which helps to better capture the global features of the matrix data and recover the missing entries.

Our MC-Weather scheme does not depend on the underlying reconstruction approach. We choose the singular value thresholding approach [65] to reconstruct the matrix.

3.2 Problem Formulation

We propose an innovative and adaptive data gathering scheme, MC-Weather, which exploits matrix completion technique and information learnt from existing data to continuously and efficiently collect weather data according to the environmental conditions. Our goal is to efficiently schedule the data collection process to significantly reduce the sensing resources needed while maintaining the sensing quality.

For N weather sensors randomly scattered in a given area, instead of letting each sensor to periodically collect and report data to the sink, in each time slot, only a subset of sensors are scheduled to perform the sensing and reporting functions based on the matrix reconstruction requirement. We define a matrix $X_{N\times T}(t)$ to hold the weather data collected within a T-slot time measurement window starting from the time slot t. In the weather matrix, a row corresponds to a sensing location and a column corresponds to a time slot. An entry represents the weather data on a particular location and time slot. The first column in the weather matrix of $X_{N\times T}(t)$ represents the weather data collected in the time slot t.

Collecting the weather information in all locations and time slots is costly. Since weather data normally have strong correlation between neighboring locations and time slots, the weather matrix should have low rank. This is confirmed with our analyses on measurement data in the next section. MC-Weather collects the weather data only at a subset of the locations in a given time slot and varies the data collection locations in different time slots. Rather than randomly selecting the measurement locations as instructed by conventional matrix completion theory, we find that the performance can be improved if we could select the collection locations more intelligently based on the information learnt from existing measurement data.

We use a Binary Sample Vector $\vec{B}(t) \in R^N$ to indicate the locations that take measurement in a given time slot t, where

$$\left[\vec{B}(t)\right]_i = \begin{cases} 1 & \text{if location } i \text{ at time } t \text{ is sampled} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Accordingly, a Binary Sampling Matrix $B_{N\times T}(t)$ can be defined as $B_{N\times T}(t) = [\vec{B}(t), \vec{B}(t+1), \ldots, \vec{B}(t+T-1)]$, and the incomplete sensory matrix $M_{N\times T}(t)$ is represented as

$$M_{N\times T}(t) = X_{N\times T}(t) \bullet B_{N\times T}(t), \tag{6}$$

where $\bullet$ represents a scalar product (or dot product) of two matrices, $M_{ij}(t) = X_{ij}(t)B_{ij}(t)$.

According to the matrix completion technique introduced in Section 3.1, when the number of samples is sufficient, the weather matrix $X_{N\times T}(t)$ can be recovered from sensory matrix $M_{N\times T}(t)$ by solving the following problem

$$\begin{aligned} &\min\ \|X(t)\|_* \\ &\text{subject to } X_{ij}(t) = M_{ij}(t) \\ &\quad M_{N\times T}(t) = X_{N\times T}(t) \bullet B_{N\times T}(t). \end{aligned} \tag{7}$$

We denote the matrix reconstructed from (7) as $\hat{X}_{N\times T}(t)$. Obviously $B_{N\times T}(t)$ directly reflects the sensing scheduling, and the key problem in our MC-Weather scheme is to identify the optimal $B_{N\times T}(t)$ $(t \geq 0)$ so as to minimize the communication cost and sensing cost while satisfying the matrix reconstruction requirement. The sampling matrix

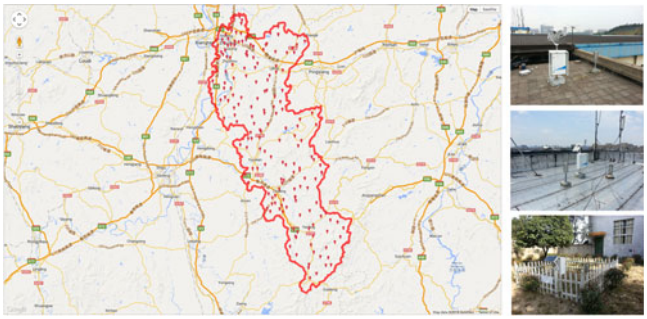


Fig. 1. Weather sensor deployment in Zhu Zhou, China.

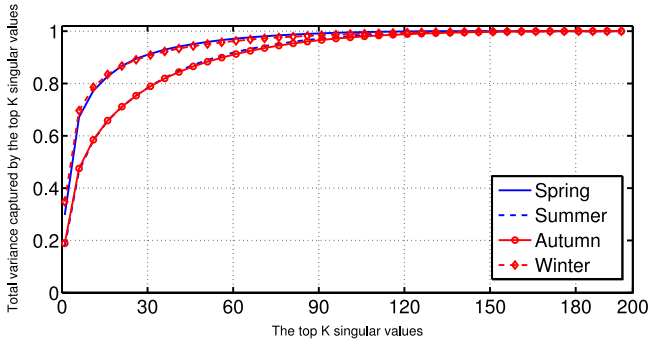


Fig. 2. Fraction captured by top k singular values.

$B_{N\times T}(t)$ indicates which locations need to take samples in a time slot.

Although the literature work on matrix completion provides some solutions to recovering data with a limited number of samples, existing schemes mostly assume that the rank of the sensory matrix is low and has a constant value. However, the weather data values (and accordingly the matrix rank) may vary significantly over time and locations, and the sparsity level (rank-level) is often not known a priori. It is thus very challenging to apply the matrix completion theory in a practical weather gathering system.

Before we present our data collection algorithm based on Intelligent Matrix Completion in Section 5, we first analyze a large set of data for weather monitoring to better understand the structure and characteristics of weather data in the next section.

4 Empirical Study with Real Weather Data

We have deployed 196 sensors to collect the weather data in Zhu Zhou, China. Fig. 1 shows the map of Zhu Zhou, where the red dot represents the location of a deployed sensor. Each sensor reports its data once an hour to the weather monitoring center via the cellular network. We have collected a large amount of weather trace data from ZhuZhou. Each data element includes weather data of rain, temperature, and wind. Specially, we choose rain data to analyze because ZhuZhou is in the area prone to flood. The trace data are collected in the duration of more than two years from 2011 to 2013. In our experiment, we set $N = 196$, $T = 168$. The trace data reveal the existence of some special structures.

4.1 Low-Rank Feature

The low-rank feature is the prerequisite for using the matrix completion. In this section, we validate that the weather matrix data have the low rank feature.

Weather data collected over different locations and time slots are not independent. There exists inherent data redundancy. We first apply the singular value decomposition (SVD) to examine whether the matrix has a good low-rank structure. A weather matrix $X_{N\times T}$ can be decomposed as

$$X = U\Sigma V^T, \tag{8}$$

where $U = [u_1, \ldots, u_N]$ is an $N \times N$ unitary matrix, $V = [v_1, \ldots, v_T]$ is a $T \times T$ unitary matrix, and Σ is an $N \times T$ diagonal matrix with the diagonal elements (i.e., the singular values) organized in the decreasing order (i.e., $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$). The rank of a matrix X, denoted by r, is equal to the number of its non-zero singular values. In this paper, we call this rank definition as "precise rank". A matrix is low-rank if its $r \ll \min\{N, T\}$.

Although the definition of the precise rank is of high theoretical interest, it is not realistic to use this definition for the practical data. The calculation of the precise rank of the matrix is an ill-posed problem in a practical environment because arbitrary small perturbations of matrix elements may change the rank [66]. Instead of performing the matrix completion based on the precise rank, this paper adopts the approximate rank [66]. We say that X has ω-rank k if

$$\inf\{\|X - Y\| : Y \quad has \quad rank \quad k\} \leq \omega. \tag{9}$$

A theorem proof provided by Eckart and Young [67] shows that the error in approximating a matrix X by X_k can be written as: $\|X - X_k\|_F^2 \leq \|X - Y\|_F^2$ where Y is any matrix with rank k, X_k is the rank-k truncated SVD of matrix X, that is $X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. The ratio $g(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{r} \sigma_i^2$ indicates what fraction of the total variance (Frobenius norm) in X is explained by the rank k approximation of X, i.e., X_k. According to Principal components analysis (PCA), if a matrix has low-rank, its top k singular values occupy the total variance, that is, $\sum_{i=1}^{k} \sigma_i^2 \approx \sum_{i=1}^{r} \sigma_i^2$.

Fig. 2 plots the fraction of the total variance captured by the top k singular values for different weather trace data from different seasons. We find that the top 20 singular values capture 70-90 percent variance in the real traces. These results indicate that the data matrix X has a good low-rank approximation in all the scenarios under investigation.

4.2 Temporal Stability

Weather data usually change slowly over time. To study the short-term stability of weather matrix, we calculate the gap between each pair of adjacent readings at a location. Specifically, the gap between each pair of adjacent readings captured in two consecutive time slots (j, and $j-1$) is equal to

$$gap(i, j) = \left|x_{ij} - x_{i,j-1}\right|, \tag{10}$$

where $1 \leqslant i \leqslant N$ and $2 \leqslant j \leqslant T$. Obviously, $gap(i, j) = 0$ if the weather data at location i is not changed from time slot $j-1$ to j. The smaller the $gap(i, j)$, the more stable the sensory readings for location i around the time slot j.

By computing the normalized difference values between adjacent time slots, we measure the temporal stability at node i and time slot j according to

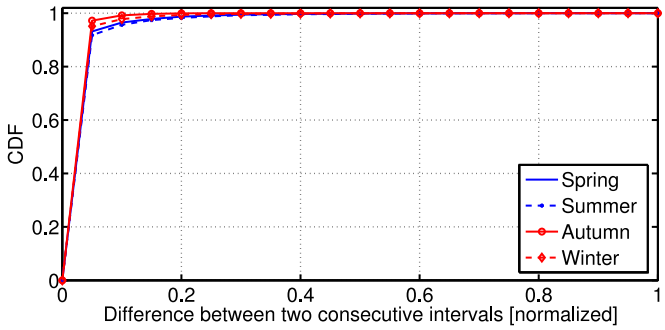


Fig. 3. Temporal stability feature.

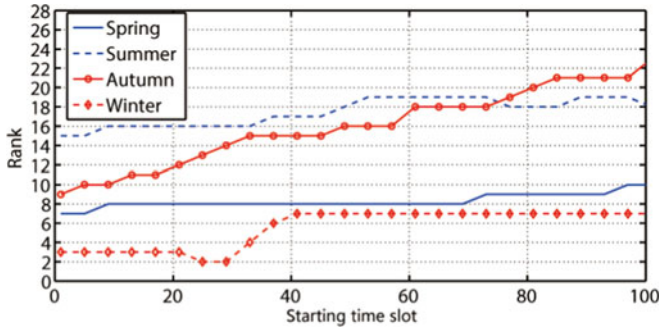


Fig. 4. Rank feature of weather data.

$$\Delta gap(i,j) = \frac{\left|x_{ij} - x_{i,j-1}\right|}{\max_{1 \leqslant i \leqslant N, 2 \leqslant j \leqslant T} \left|x_{ij} - x_{i,j-1}\right|}, \qquad (11)$$

where $\max_{1 \leqslant i \leqslant N, 2 \leqslant j \leqslant T} \left|x_{ij} - x_{i,j-1}\right|$ is the maximal gap between any two consecutive time slots in the weather matrix.

We plot the CDF of $\Delta gap(i,j)$ in Fig. 3. The X-axis represents the normalized difference values between two consecutive time slots, i.e., $\Delta gap(i,j)$. The Y-axis represents the cumulative probability. We observe that more than 90 percent $\Delta gap(i,j)$ are very small (< 0.05). These results indicate that temporal stability exists in real environments. In Section 5.4.2, we design our cross sample model by utilizing this feature.

4.3 Rank-Stability

We plot the rank of the consecutive weather matrix in Fig. 4 by varying the starting time slot from 0 to 100 to further investigate the rank feature. In this paper, we calculate the matrix rank according to (9) by setting $\omega = 0.01$. Each weather matrix only includes the sensing data of T time slots. The X-axis represents the first time slot of a weather matrix. The Y-axis represents the matrix's rank of the corresponding T-time-slot measurement window.

Obviously, the weather matrix does not have a constant rank and the rank of matrix varies with time slots and seasons, which contradicts to the assumption in existing work that the matrix has the constant rank. On the other hand, even though the rank of weather matrix may change, the rank between adjacent matrices changes only slightly, thus there exists relative rank stability. In Section 5.2, we will exploit the relative rank stability in our learning algorithm for more efficient on-line weather gathering.

5 On-Line Weather Gathering Based on Matrix Completion

In this section, by taking advantage of the weather matrix's low-rank, temporal stability, and relative rank stability features, we design an innovative on-line weather gathering scheme (MC-Weather) based on matrix completion to efficiently schedule the data collection at different sensors for lower sensory cost while ensuring accurate $X_{N \times T}$ reconstruction. Compared to sampling at each location and time slot, this leads to a variety of benefits, including low power consumption, long lifespan of sensors, and reduced data transmissions in the network.

5.1 Rank of Adjacent Matrices

To support continuous weather gathering and reduce the computation cost for reconstructing the weather matrix, our

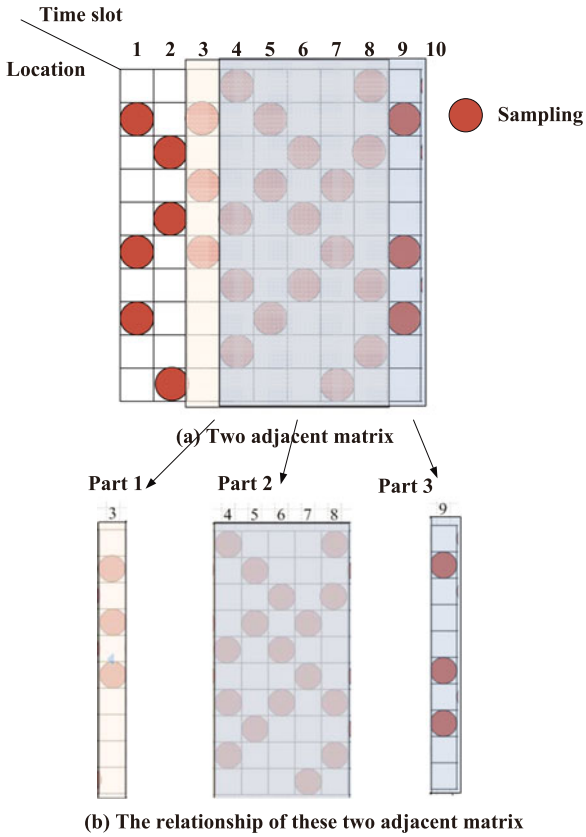


Fig. 5. Slide window based weather gathering.

MC-Weather is implemented based on the sliding window model, where the oldest time slot in the window is removed when a new time-slot is added to the window. We apply matrix completion technique to reconstruct the weather matrix from the sensory matrix obtained in a window, and we call the window containing the current time slot the *active measurement window*.

Fig. 5 shows an example sliding window with the size $T = 6$, and the current time slot is 9. There are two adjacent measurement windows, with the first one from time slots 3 to 8 and the second one from 4 to 9. The second one is the active measurement window. There are 10 sensors in the system, and the two adjacent weather matrices corresponding to these two windows are denoted by $X_{10\times6}(3)$ and $X_{10\times6}(4)$.

From the matrix completion theory, the rank of the matrix has a direct impact on the number of samples required to accurately reconstruct the weather matrix from partial sensory data. In a dynamic environment, however, it

is difficult to determine the number of samples needed in a new active window because the rank of its corresponding matrix is unknown.

As shown in Fig. 5b, obviously, most columns of the two adjacent matrices are the same except one column. Therefore, there exists a strong relationship between these two matrices. Before we discuss their relationship in Theorem 2, the following Theorem presents the rank relationship of two matrices with the same number of rows. Let (A, B) be a matrix formed with A and B concatenated.

Theorem 1. *Given two matrices $A \in R^{m\times n}$ and $B \in R^{m\times k}$, the rank of matrix A, B and (A, B) satisfies*

$$\max\{rank(A), rank(B)\} \leqslant rank(A, B) \leqslant rank(A) + rank(B). \tag{12}$$

Specially, if B is a non-vanishing vector and $B \in R^m$, we have

$$rank(A) \leqslant rank(A, B) \leqslant rank(A) + 1. \tag{13}$$

Proof. The rank of a matrix is the number of dimensions of the space spanned by it. Thus, by concatenating two matrices (A, B), one extreme case is that the resulting space will be unchanged (e.g., each column of A is a linear combination of columns of B) and the other is that their rows/columns are linearly independent and thus the number of dimensions will be added. Therefore, we have (12).

If B is a non-vanishing vector in R^m, obviously $rank(B) = 1$ and we can obtain

$$rank(A) \leqslant rank(A, B) \leqslant rank(A) + 1, \tag{14}$$

which completes the proof. □

Theorem 2. *Given two weather matrices of adjacent windows $X_{N\times T}(t), X_{N\times T}(t+1)$ and $rank(X_{N\times T}(t)) = r$, the rank of the matrix $X_{N\times T}(t+1)$ satisfies*

$$r - 1 \leqslant rank(X_{N\times T}(t+1)) \leqslant r + 1. \tag{15}$$

Proof. Because $X_{N\times T}(t), X_{N\times T}(t+1)$ are the weather matrices of adjacent windows, we can obtain that $X_{N\times T}(t) = (\vec{B}(t), X_{N\times(T-1)}(t+1))$ and $X_{N\times T}(t+1) = (X_{N\times(T-1)}(t+1), \vec{B}(t+T-1))$ where $\vec{B}(t) \in R^N$ and $\vec{B}(t+T-1) \in R^N$ are non-vanishing vectors.

Calculating the rank of $X_{N\times T}(t)$ by applying Theorem 1, we can obtain

$$\begin{aligned} rank\big(X_{N\times(T-1)}(t+1)\big) &\leqslant rank(X_{N\times T}(t)) \\ &\leqslant rank\big(X_{N\times(T-1)}(t+1)\big) + 1. \end{aligned} \tag{16}$$

From (16), we can further obtain

$$r - 1 \leqslant rank\big(X_{N\times(T-1)}(t+1)\big) \leqslant r. \tag{17}$$

Calculating the rank of $X_{N\times T}(t+1)$ by applying Theorem 1, we can obtain

$$\begin{aligned} rank\big(X_{N\times(T-1)}(t+1)\big) &\leqslant rank(X_{N\times T}(t+1)) \\ &\leqslant rank\big(X_{N\times(T-1)}(t+1)\big) + 1. \end{aligned} \tag{18}$$

Combining (17) and (18), we can obtain

$$r - 1 \leqslant rank(X_{N\times T}(t+1)) \leqslant r + 1, \tag{19}$$

which completes the proof. □

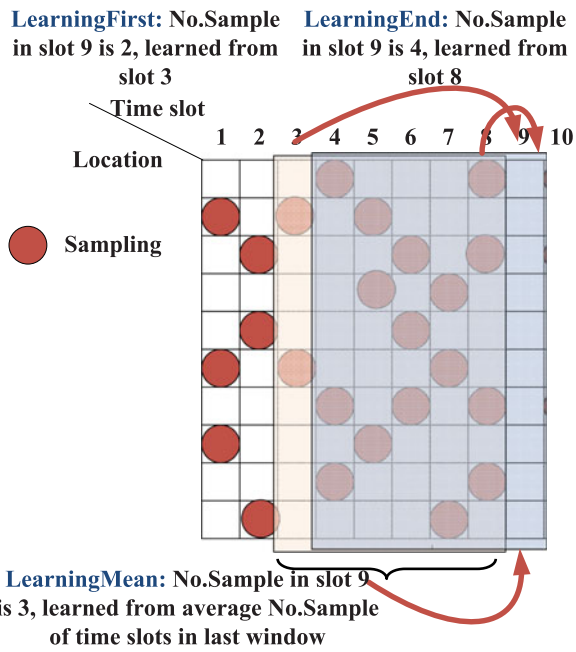


Fig. 6. Different sample learning principle.

Theorem 2 verifies the relative rank stability feature which we have observed from real weather data traces in Section 4. Based on this feature, we will design our learning-based scheduling scheme for sensor data collection in the following section.

5.2 Sample Learning Principle

As proven in Theorem 2, the rank difference of adjacent weather matrices is no more than 1. Based on this feature, the number of samples to take in a new time slot t can be learnt from the last window. Accordingly, we propose three learning principles to identify the initial sampling number to use in the new time slot:

- *LearningFirst*. The number of samples to take in a new time slot t is learnt and set to the same as that in the time slot $t - T$. In Fig. 6, the initial number of samples to take in slot 9 is set to 2, the same as that in slot 3.
- *LearningEnd*. The number of samples to take in a new time slot t is learnt and set to the same as that in the time slot $t - 1$. In Fig. 6, the initial sampling number in slot 9 is learnt from slot 8 and set to 4.
- *LearningMean*. The number of samples to take in a new time slot t is learnt and set to be the average sampling number of those from time-slots in the last window. In Fig. 6, the initial sampling number in slot 9 is set to 3.

Obviously, if two adjacent windows have the same rank, *LearningFirst* is the most effective principle. As shown in Fig. 6, there are three parts in the two continuous measurement windows, part1 (slot 3), part 2 (slots 4, 5, 6, 7, 8) and part 3 (slot 9). When the time slot 9 starts, the samples in part 1 and part 2 remain the same. If the number of samples in the previous measurement window ranging from slots 3 to 8 are sufficient to reconstruct the weather data with high reconstruction accuracy, it is also sufficient to set the number of samples in slot 9 to 2, the same as that in slot 3. If two adjacent windows

have different ranks and the ranks vary with time, the last time slot can better approximate the rank from the previous window, therefore, LearningEnd may be more effective.

In a practical data gathering process, the sink node can apply a learning principle according to the environmental conditions. In the simulation part, we will compare the performance of different learning principles.

5.3 Adaptive Sampling

This paper focuses on continuous and on-line data gathering in WSNs. As it is impossible to know the whole matrix data a priori and only the sample data are observed, one challenging problem is how we can determine whether the matrix has been accurately reconstructed in the sequential sampling process.

In this section, we first present the opportunity to identify whether the whole matrix is recovered given only the sample data, then propose our adaptive sampling algorithm.

5.3.1 Reconstruction Error Control

Given a matrix A and the sample index set Ω, we let $\tilde{A}$ to denote the recovered matrix through the matrix completion using the samples in Ω. The reconstruction error on samples is defined as $\frac{\|P_\Omega(\tilde{A}-A)\|_F}{\|P_\Omega(A)\|_F}$. The reconstruction error on the whole matrix is defined as $\frac{\|\tilde{A}-A\|_F}{\|A\|_F}$.

Following we will show that given a matrix A with only entries in the index set Ω known, the reconstruction error of the whole matrix can be controlled though the control of the reconstruction error on samples.

From [65], for a fixed matrix $A \in R^{n_1 \times n_2}$ under the incoherence assumption in [25], we have the following relationship with a very large probability

$$(1-\epsilon)\frac{m}{n_1 \times n_2}\|A\|_F^2 \leq \|P_\Omega(A)\|_F^2 \leq (1+\epsilon)\frac{m}{n_1 \times n_2}\|A\|_F^2, \quad (20)$$

provided that the rank of A is not too large, where ϵ is a constant that is smaller than $1/2$ [65]. Based on Eq. (20), we have

$$\|P_\Omega(A)\|_F^2 = c_1 \frac{m}{n_1 \times n_2}\|A\|_F^2, \quad (21)$$

where c_1 is a small constant in the range of $\frac{1}{2}$ to $\frac{3}{2}$. Applying (20) to the matrix $\tilde{A} - A$, we have

$$\left\|P_\Omega(\tilde{A}-A)\right\|_F^2 = c_2 \frac{m}{n_1 \times n_2}\left\|\tilde{A}-A\right\|_F^2, \quad (22)$$

where c_2 is a small constant in the range of $\frac{1}{2}$ to $\frac{3}{2}$. Combining (21) and (22), we have

$$\frac{\left\|P_\Omega(\tilde{A}-A)\right\|_F^2}{\|P_\Omega(A)\|_F^2} = c_3 \frac{\left\|\tilde{A}-A\right\|_F^2}{\|A\|_F^2}, \quad (23)$$

where $c_3 = \frac{c_2}{c_1}$ is a constant with the value in the range of $\frac{1}{3}$ to 3. As the value of $\frac{\|P_\Omega(\tilde{A}-A)\|_F^2}{\|P_\Omega(A)\|_F^2}$ is proportional to the value of $\frac{\|\tilde{A}-A\|_F^2}{\|A\|_F^2}$, we could control the reconstruction error of the whole matrix by controlling the relative error at the set of sampled locations.

5.3.2 The Proposed Algorithm

We propose our adaptive sampling algorithm in Algorithm 1. In step 1, the sampling number in a new time slot t, C, is determined following the learning principle of choice. With C new samples taken in the slot t, the sink runs the matrix reconstruction algorithm to obtain data in the active window $\hat{X}_{N\times T}(t-T+1)$ and calculate the reconstruction error ε according to Eq. (24), which is the reconstruction error on samples with $B_{ij}(t) = 1$.

Algorithm 1. The Matrix Completion Based Adaptive Sampling Algorithm

1: Based on a learning principle selected, identify the initial sampling number to use in the new time slot t, denoted as C. According to the cross sampling principle in Section 5.4.2, select sampling locations and initialize $\vec{B}(t)$ with $|\vec{B}(t)| = C$. The sink announces the sampling schedule according to $\vec{B}(t)$.
2: Once receiving C measurements, the sink runs the matrix reconstruction algorithm to obtain data in the active window $\hat{X}_{N\times T}(t-T+1)$ and calculate the reconstruction error ε as

$$\varepsilon = \frac{\sqrt{\sum_{i,j,B_{ij}(t)=1}\left(M_{ij}(t)-\hat{X}_{ij}(t)\right)^2}}{\sqrt{\sum_{i,j,B_{ij}(t)=1} M_{ij}(t)^2}} \quad (24)$$

3: **while** $|\varepsilon - \varepsilon_b| > \beta$ **do**
4: **if** $\varepsilon - \varepsilon_b > 0$ **then**
5: Add $\alpha C(\varepsilon - \varepsilon_b)$ extra measurements according to cross-based sampling principle in Section 5.4.2, and update $\vec{B}(t)$ and $C = C + \alpha C(\varepsilon - \varepsilon_b)$.
6: **else**
7: Obtain the effective number of samples in the time slot t by updating $C = C - \alpha C(\varepsilon_b - \varepsilon)$ and $\vec{B}(t)$.
8: **end if**
9: Based on the updated $\vec{B}(t)$, calculate the reconstruction error ε according to Eq. (24).
10: **end while**
11: The sink stores $\vec{B}(t)$ to indicate the effective sampling in time slot t.

If the recovery error is large, our algorithm goes to the step 5 to determine the number of supplemental measurements to take. Without knowing the actual number of samples needed, the sink could schedule sensors to take additional samples in multiple rounds until the recovery accuracy is reached. A straight-forward approach is to take additional samples at a given rate in each round at the cost of extra computational and communication cost. To reduce the overhead, we propose to adapt the sampling number according to the recovery error ε and the tolerable error ε_b. We add $\alpha C(\varepsilon - \varepsilon_b)$ extra measurements according to cross-based sampling principle to be presented in Section 5.4.2, and update $C = C + \alpha C(\varepsilon - \varepsilon_b)$ until the error gap $\varepsilon - \varepsilon_b$ is smaller than β.

If the error is too low with $\varepsilon_b - \varepsilon > \beta$, the reconstruction can reach the accuracy requirement, and we consider $\hat{X}$ as a successful recovery. However, we don't expect to take too many samples unnecessarily in the future. To get a proper sample number that can guide sampling in future slots, our algorithm goes to step 7 to obtain the effective number of

samples in the current time slot by updating $C = C - \alpha C(\varepsilon_b - \varepsilon)$ until the error gap $(\varepsilon_b - \varepsilon)$ is smaller than β.

When the updating process above stops, the resulting C is the number of effective samples needed in the time slot t. According to the learning principles proposed in Section 5.2, the number of effective samples in current time slot will guide the future weather monitoring.

5.4 Sampling Initiation and Scheduling

Our adaptive sampling algorithm provides a guide on the number of samples to take in a new time slot based on the information from the previous measurement window and the recovery error. However, at the beginning of the data gathering procedure, there are not enough historical measurements to guide the sampling process. We introduce a training phase in Section 5.4.1 to initialize the sampling process based on data collected from the first T-time slots, and a scheme to determine the sampling locations in each time slot in Section 5.4.2.

5.4.1 Uniform Time-Slot Sampling

In the training phase, each sensor senses and reports data to the sink. The key problem to solve in this phase is to identify the effective sampling set among all measurement data to initialize the sampling schedule for future time slots.

As all locations are sensed in the training phase, the sink knows the exact weather data $X_{N\times T}(1)$ and the rank of $r = rank(X_{N\times T}(1))$. Therefore, the sink can infer the effective sampling number m according to Eq. (4).

Obviously, the sample distribution has direct impact on the reconstruction accuracy. To reconstruct the matrix, the samples should be taken randomly to avoid matrix completion failure when a row or a column is un-sampled.

In [68], the authors analyze two models to obtain the sample set, the Bernoulli model and the uniform model. Under the Bernoulli model, each entry in the matrix is sampled with a probability $p = m/(n_1 \times n_2)$ (where n_1 and n_2 are the number of rows and columns of the matrix, respectively). Under the uniform model, Ω is taken uniformly at random from the matrix with the cardinality of Ω being m. The two models were shown to have the equivalent performance.

In our adaptive sampling algorithm, the samples taken in a time slot t can guide the sample-taking process in future time slots. If applying the uniform model or the Bernoulli model, we cannot guarantee that every time slot has samples. When there is no sample in a column, we cannot know the number of samples to take in later time slots. Neither of the existing sample models is suitable to apply in our MC-weather gathering scheme. We propose our uniform time slot sampling model as follows.

The desired sampling model in MC-weather gathering scheme should be simple to implement, and have an equal number of samples in each time slot in the training window so that every time slot has sampling data and can reflect the rank of the training window. Accordingly, we propose a uniform time-slot sampling model so that the number of samples taken in each time slot within the training window is equal and set to $\lceil \frac{m}{T} \rceil$.

With the number of samples to take in each column determined, we still need to identify the locations to take samples in each time slot. In the following section, we propose our cross sampling principle to achieve this goal.

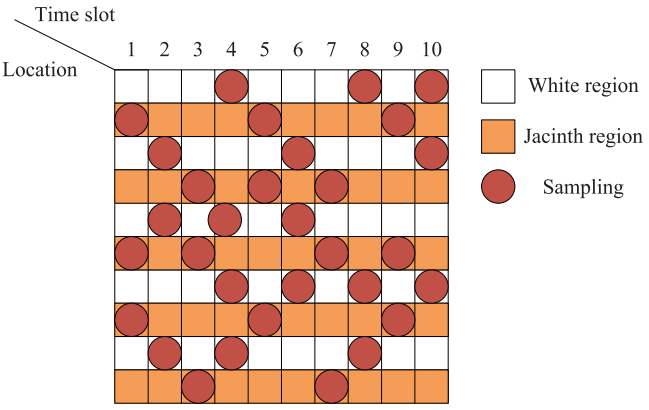


Fig. 7. UTSCS sampling model.

5.4.2 Cross Sampling Principle

Due to the temporal stability of sampling data, the desired sampling principle in MC-Weather scheme should avoid sampling the same location in adjacent time slots. To achieve the objective, we divide the locations into two parts, and different time slots have different priority to sample one of the parts. We call this *cross sampling principle*.

Fig. 7 shows an example of our Uniform Time-slot and cross-based sampling model. The training measurement window is $X_{10\times 10}(1)$ and the required sampling number is $m = 30$. According to the model, each time slot should have 30/10 = 3 effective samples. Moreover, to implement uniform sampling while not sampling the same location in adjacent time slots to increase the data diversity, the locations are divided into two parts, white part and jacinth part. In the time slot 1, effective samples have high priority to take in jacinth part, while in the time slot 2, effective samples have higher priority to take in the white part.

5.4.3 Sample Model Analysis

The key difference between our Uniform Time-Slot and Cross based Sampling model and the other two models (Bernoulli model and the uniform model) is that under the UTSCS model, every column is guaranteed to be sampled at least once and the same location is avoided to take samples in adjacent time slots. Although UTSCS adopted in this paper divides the locations into two parts and schedules the sampling based on the partitions, our UTSCS can be easily extended to support more partitions for more uniform sampling.

Fig. 8 shows an example of UTSCS in which locations are divided into four parts (Jacinth region, Purple region, Blue region, White region). Totally 24 samples are distributed into 12 time slots, thus each time slot has two samples. Similar to Fig. 7, in the time slot 1, samples have a high priority to take in the white part; in time slot 2, samples are taken in the jacinth part; in the time slot 3, samples are taken in the purple part; in the time slot 4, samples are taken in the blue part.

It is clear that if we fail to observe at least one entry in a row (or a column) of the matrix, we have no way of recovering the matrix. In the Theorem 3 below, we will show that the probability of missing an entire row under our UTSCS model decreases with the increase of k.

Theorem 3. *Let F be the event that an entire row is missed to sample. The probability of event F (denoted by $P_{UTSCS}(k)$)*

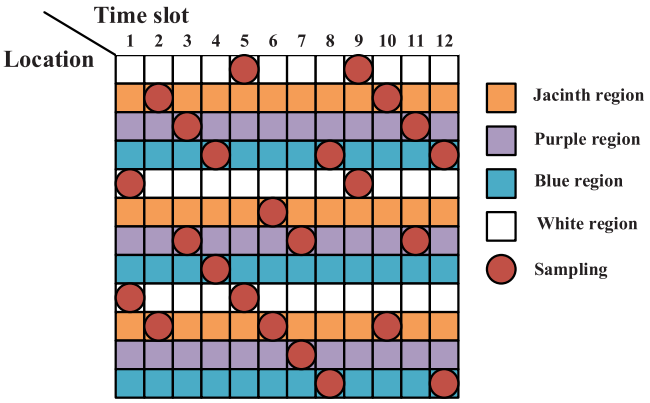


Fig. 8. UTSCS sampling model with k=4.

under UTSCS sampling model satisfies that

$$\begin{cases} P_{UTSCS}(k) > P_{UTSCS}(k+1) & k \le \frac{NT}{m+T} \\ P_{UTSCS}(k) = 0 & k > \frac{NT}{m+T}, \end{cases} \tag{25}$$

where k is the number of partitions.

Proof. Our UTSCS adopts a uniform time-slot sampling model. Under such a model, when there are m samples, each time slot has m/T samples. If we divide the N sensor locations into k parts, in one time slot, there are N/k locations to schedule the sampling in each part.

Under the UTSCS sampling model, the probability of event F is

$$P_{UTSCS}(k) = \left(\frac{C_{m/T}^{N/k-1}}{C_{m/T}^{N/k}}\right)^{T/k} = \left(\frac{NT-km}{NT}\right)^{T/k}, \tag{26}$$

where $\left(\frac{C_{m/T}^{N/k-1}}{C_{m/T}^{N/k}}\right)$ is the probability that an entry is not sampled. In (26), $C_{m/T}^{N/k-1}$ is the combination of selecting m/T locations among $N/k-1$ locations to take sample. $C_{m/T}^{N/k}$ is the combination of selecting m/T locations among N/k locations to take sample. The exponent in (26) is T/k because when there are totally k parts, each part should be sampled within T/k time slots.

Obviously, such a probability $P_{UTSCS}(k)$ holds when $N/k \ge m/T$ and $N/k-1 \ge m/T$ (that is $k \le \frac{NT}{m+T}$) are satisfied. Let $\log\left(P_{UTSCS}(k)\right) = \frac{T}{k}\log\left(\frac{NT-km}{NT}\right)$. We have

$$\begin{aligned} &\frac{\partial \log\left(P_{UTSCS}(k)\right)}{\partial k} \\ &= -\frac{T}{k^2}\log\left(\frac{NT-km}{NT}\right) + \frac{1}{\ln 10}\times\left(\frac{NT-km}{NT}\right)^{-1}\times\frac{-m}{NT}\times\frac{T}{k} \\ &= \frac{-T}{k}\left(\frac{1}{k}\times\frac{1}{\ln 10}\times\left(-\ln\left(\frac{NT}{NT-km}\right)\right)+\frac{1}{\ln 10}\right. \\ &\quad \left.\times\left(\frac{NT}{NT-km}\right)\times\frac{m}{NT}\right) \\ &= \frac{-T}{k}\times\frac{1}{\ln 10}\left(\frac{1}{k}\times\left(-\ln\left(\frac{NT}{NT-km}\right)\right)+\frac{m}{NT-km}\right) \\ &= \frac{-T}{k}\times\frac{1}{\ln 10}\times\frac{1}{k}\left(-\ln\left(\frac{NT}{NT-km}\right)+\frac{km}{NT-km}\right) \\ &= \frac{T}{k}\times\frac{1}{\ln 10}\times\frac{1}{k}\left(\ln\left(\frac{NT}{NT-km}\right)+1-\frac{NT}{NT-km}\right). \end{aligned} \tag{27}$$

Let $\theta = \frac{NT}{NT-km}$, we obtain $\frac{\partial \log\left(P_{UTSCS}(k)\right)}{\partial k} = \frac{T}{k}\times\frac{1}{\ln 10}\times\frac{1}{k}(\ln\theta+1-\theta)$. We can see that $f(\theta) = (\ln\theta+1-\theta)$ is a continuous function of θ when $\theta > 0$. Further, $f(\theta)$ is an decreasing function with θ when $\theta > 1$.

Obviously, we have $NT > NT-km$, and thus $\theta > 1$. Therefore, $f(\theta) < f(1) = (\ln 1+1-1) = 0$, based on which we have that $\frac{\partial \log\left(P_{UTSCS}(k)\right)}{\partial k} < 0$ and $P_{UTSCS}(k)$ is a decreasing function with k when $k \le \frac{NT}{m+T}$.

When $N/k-1 < m/T$ thus $k > \frac{NT}{m+T}$, under the UTSCS sampling model, no row will miss sample, that is, we have $P_{UTSCS}(k) = 0$. Therefore, we have (25) and the proof completes. □

Although Theorem 3 shows that more partitions bring a smaller probability of missing an entire row, it also brings a larger complexity in scheduling the sample taking. For practical and simple implementation, this paper adopts UTSCS with $k = 2$ in the performance study.

From [68], we know that sampling according to Bernoulli model has been analyzed and shown to be able to recover the matrix satisfactorily with high probability. Under the Bernoulli model, as each sample is taken independently, the probability of event F is $P_{Bernoulli} = \left(1-\frac{m}{NT}\right)^T = \left(\frac{NT-m}{NT}\right)^T$, where $\frac{m}{NT}$ is the probability that each entry in the matrix is sampled. Obviously, $P_{Bernoulli} = P_{UTSCS}(1) = \left(\frac{NT-m}{NT}\right)^T$ when $k = 1$, that is, Bernoulli model is a special case under UTSCS.

As Bernoulli model is a special case under UTSCS when $k = 1$, according to (25), the sampling using our UTSCS (with $k \ge 2$) has the probability of missing an entire row smaller than that of using the Bernoulli model. In the simulation part, we will show that our UTSCS with $k = 2$ can achieve better performance for matrix completion compared with the sampling based on Bernoulli model.

5.5 Complete MC-Weather Gathering Scheme

As shown in Fig. 9, the whole MC-weather gathering scheme can be summarized as follows.

(1) In the training phase at the beginning of the first T time slots, every node senses and sends weather data to the sink (Fig. 9a).
(2) The Uniform Time-slot and cross Sampling model is applied to identify the effective sample sets within the training window (Fig. 9b).
(3) In each new time slot t after $(T-1)$th time slot, the sink node first identifies the initial sample number following the proposed sample learning principle, and then identifies an initial sample set in this new time slot according to the cross sampling principle (Fig. 9c).
(4) The sink can adapt the sampling set following the adaptive algorithm in Section 5.3 to accurately reconstruct the weather matrix in the presence of the change of environmental conditions and accordingly the rank of the weather data matrix (Figs. 9d and 9e).

6 Performance Evaluations

To evaluate the performance of our MC-Weather algorithm, we first perform simulations using weather traces, then we check two existing environment trace data (e.g., PM2.5 and

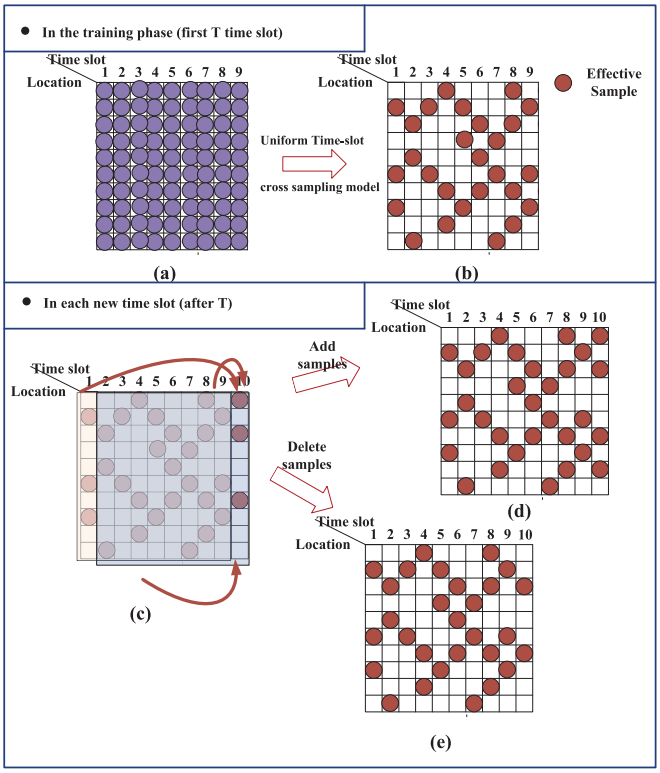


Fig. 9. MC-Weather gathering scheme.

PM10) to validate the features and also evaluate the performance of our MC-Weather algorithm.

6.1 Evaluations on Weather Traces

We first perform extensive simulations based on data traces collected by our deployed 196 weather sensors. Specifically, we chose the rain traces gathered from July 1 to August 31st, 2012.

6.1.1 Impact of α and β

According to Algorithm 1, the number of samples to add or reduce in each new time slot is adaptively set to $\alpha C(\varepsilon_b - \varepsilon)$ until the reconstruction error ε satisfies $|\varepsilon - \varepsilon_b| \leq \beta$, where ε_b and β are the tolerable error and the allowable error deviation, respectively. Usually, we can set the tolerable error ε_b and the allowable error deviation β according to the requirement of applications. In this paper, to control the reconstruction error of matrix completion in weather data gathering, the tolerable error is set to $\varepsilon_b = 0.4\%$. To investigate the impact of α and β on the convergency of our MC-Weather scheme, we vary these two parameters and run Algorithm 1.

Fig. 10 shows the convergency behavior under different α and β. It is clear that under all α and β, our MC-Weather scheme is able to converge within two iterations to get accurate reconstruction error within $\varepsilon_b - \beta \leq \varepsilon \leq \varepsilon_b + \beta$. As expected, it converges faster for larger β, as β directly impacts the stopping condition in the MC-Weather scheme. According to Fig. 10, we set $\beta = 0.05\%$ and $\alpha = 500$ in this paper.

6.1.2 Impact of Sampling Learning Principles

To evaluate the performance with different sample learning principles proposed in Section 5.3, we calculate the gap between the learned initial sample number (denoted by $C_{learning}[i]$ for time slot i) and final effective sampling number (denoted by $C_{effective}[i]$ for time slot i) obtained from the adaptive algorithm. Specifically, the gap ratio between these two sample numbers is equal to

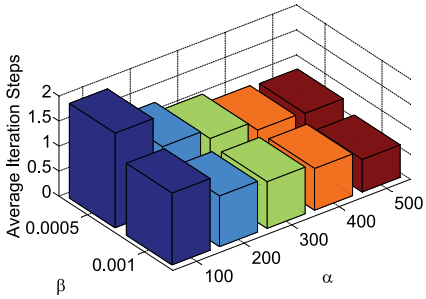


Fig. 10. Impact of α and β.

$$Gap_{ratio}[i] = \frac{|C_{Learning}[i] - C_{Effective}[i]|}{C_{Effective}[i]}. \quad (28)$$

Obviously, the smaller the resulting $Gap_{ratio}[i]$, the better the learning principle is.

We plot the CDF of $Gap_{ratio}[i]$ in Fig. 11. The X-axis presents the gap ratio. The Y-axis presents the cumulative probability. For the LearningEnd principle, more than 90 percent probability the values of $Gap_{ratio}[i]$ are very small (< 0.01). This indicates that the LearningEnd principle is more suitable to apply in data gathering when the environment is dynamic and the rank of the data matrix varies. Accordingly, we adopt the LearningEnd principle in our practical weather gathering system.

6.1.3 Performance Comparison

To evaluate the performance of our scheme, we implement four weather gathering schemes. The first scheme is our MC-Weather scheme in which the effective number of samples in the training windows are obtained according to our UTSCS Sampling model proposed in Section 5.4 and the effective samples in each new time slot is set according to Algorithm 1. According to the result of Theorem 3, we take data from one week for training purpose with the training windows set to $196 \times 168 (168 = 24 \times 7)$. The second scheme is a uniform random sample scheme. Given a fix sampling ratio, the sensors in each location take samples according to the uniform sampling model with three different sampling ratios, 0.6, 0.7 and 0.8, denoted as Uniform 0.6, Uniform 0.7, and Uniform 0.8, respectively. In the third scheme, the uniform-time slot sampling model proposed in Section 5.4.1 is applied in the training windows, while for each new time slot, the uniform sampling with a given sampling ratio (0.6) is applied, denoted as TimeUniform-0.6. Different from the third scheme, in the fourth scheme, our cross sampling principle proposed in Section 5.4.2 is applied to identify the sample set in a new time slot, denoted as TimeUniformCross-0.6.

- Estimation error

In Fig. 12, we compare the reconstruction errors of the four weather data gathering schemes. The reconstruction errors of three peer schemes fluctuate over the time period simulated, while the errors of our MC-Weather remain low and stable. This is because that the rank of the weather data

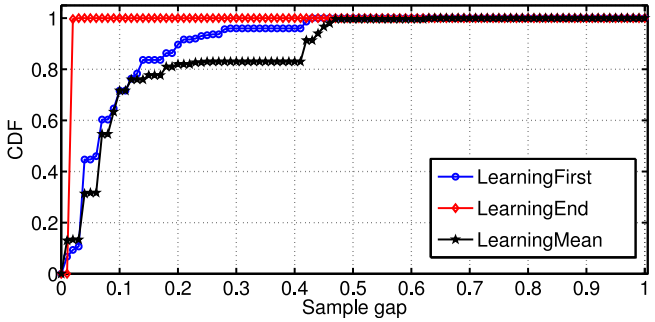


Fig. 11. Sample gap under different sample learning principle.

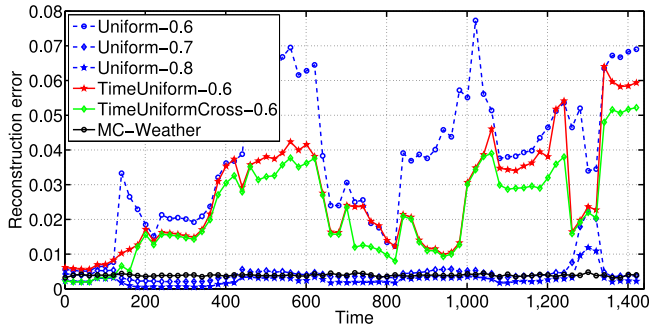


Fig. 12. Reconstruction error.

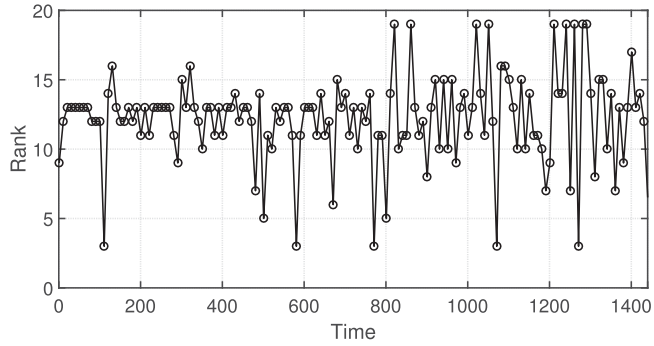


Fig. 13. Rank.

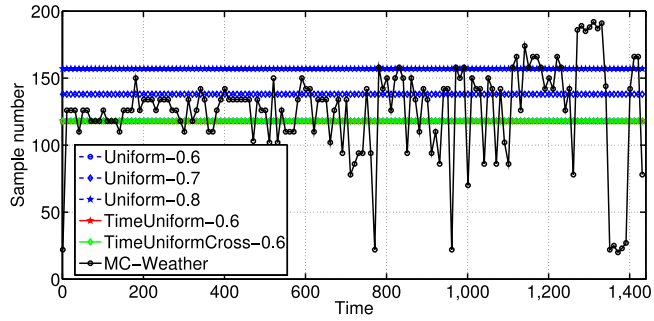


Fig. 14. Adaptive sample number.

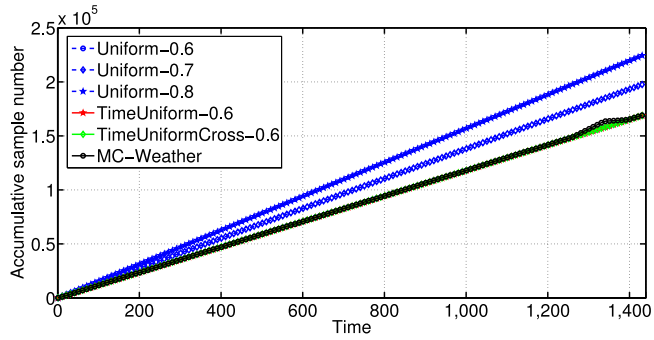


Fig. 15. Total sample number.

varies with time, and sampling with a fixed ratio is not suitable for dynamic weather data gathering.

As shown in Fig. 12a, the error rate under Uniform-0.8 around time slot 1,300 raises up to 1 percent, which is much larger than the tolerable error ($\varepsilon_b = 0.4\%$). These simulation results indicate that even for Uniform-0.8 (sampling with the largest ratio), the reconstruction error can not be controlled at a low level if the samples are simply taken randomly as done in the literature work. In contrast, the error rate of MC-Weather can be well controlled to be around the tolerable error during the testing period, which demonstrates that MC-weather can successfully adapt the sampling rate in response to the change of data in a dynamic environment.

Compared to Uniform-0.6 scheme, TimeUniform-0.6 has lower error even though both schemes have the same sampling ratio. This implies that the sampling model taken by TimeUniform-0.6 is better to apply with the matrix completion to recover data. Moreover, compared to TimeUniform-0.6, the TimeUniformCross-0.6 has lower error rate by following our cross sampling principle to avoid taking samples from the same location in adjacent time slots. These results demonstrate that our uniform time slot and cross sampling model help to achieve much better data gathering performance.

- Sample number

Figs. 14 and 15 compare the number of samples and the accumulative number of samples taken under different schemes. To better understand the effectiveness of the adaptive sampling strategy, we also draw Fig. 13 to show the rank variation of the matrices over time. Although Fig. 4 shows that rank between adjacent matrices changes only slightly, in Fig. 13, we find in a longer period time, the rank of matrices varies over a large range, and thus the static sampling strategy with fix sampling ratio can not work well. As expected, when the rank becomes large in Fig. 13, the number of samples needed become large accordantly in Fig. 14.

In consistence with the results shown in Fig. 12, the curves in all the schemes in Fig. 14 are parallel to the X-axis except our MC-scheme. This is because the other schemes utilize a fixed sampling ratio, while MC-weather can adjust the sampling ratio according to the rank variation to accurately recover data matrix while reducing the sampling overhead.

Fig. 15 also demonstrates that our uniform Time slot and cross sample model is good for matrix completion. As shown in Fig. 15, the accumulative sample number of MC-Weather is not larger than the other three schemes (Uniform-0.6, TimeUniform-0.6, TimeUniformCross-0.6) while the error rate of MC-Weather is much smaller (Fig. 12).

6.2 Evaluations on Other Environment Traces

The information about the urban air quality, e.g., the concentrations of PM2.5 and PM 10 is of great importance to protect the human health and control the air pollution. Besides the weather traces, we use two publicly available PM2.5 and PM10 [69] to further validate the features found in the weather data and also evaluate the performance of

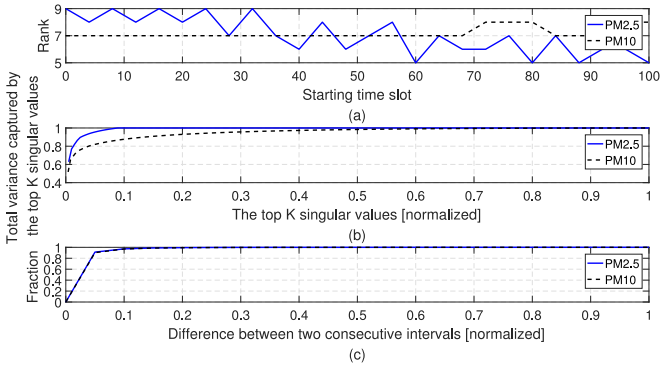


Fig. 16. Features in traces PM 2.5 and PM 10.

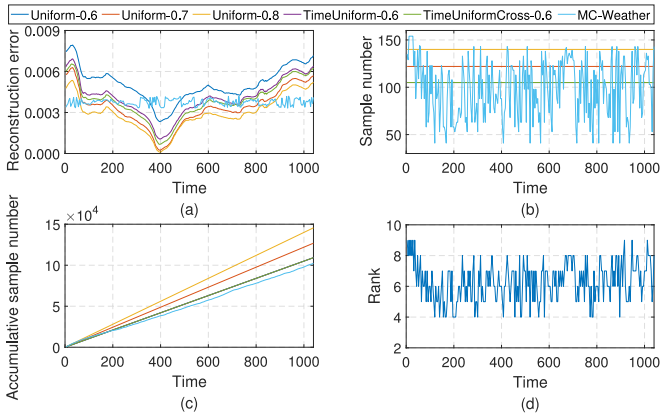


Fig. 17. Performance on trace PM 2.5.

our proposed MC-Weather in gathering the PM2.5 and PM10 data. Specifically, we chose the traces gathered from May 1 to June 15, 2013.

6.2.1 Feature Validation

In Section 4, we reveal that weather data has the features of low-rank, temporal stability, and relative rank stability. Following the similar simulation setup in Section 4, we perform the simulations on the traces PM2.5 and PM10. As almost all physical conditions monitored are continuous without sudden changes, sensory data generally exhibit strong spatio-temporal correlation [70]. Thus the sensory data matrix (PM2.5 and PM 10) has the feature of Temporal stability (in Fig. 16c) and low-rank (in Fig. 16b). We also draw the rank of the consecutive data matrix in Fig. 16a by varying the starting time slot from 0 to 300 to further investigate the rank feature. As expected, we observe that the rank between adjacent matrices changes only slightly, and thus the PM 2.5 and PM10 trace also have the feature of rank-stability.

6.2.2 Performance Comparison

PM 2.5 and PM 10 data traces have the features of low-rank, temporal stability, and relative rank stability. As these features are the basis for our design of the MC-weather scheme, we expect that our scheme can also work well to collect the air quality data with low cost. As demonstrated in Figs. 17 and 18, our MC-Weather can adaptively change the sampling ratio according to the variation of matrix ranks. Compared with other sampling strategy with a fix sampling ratio, our MC-Weather can control the

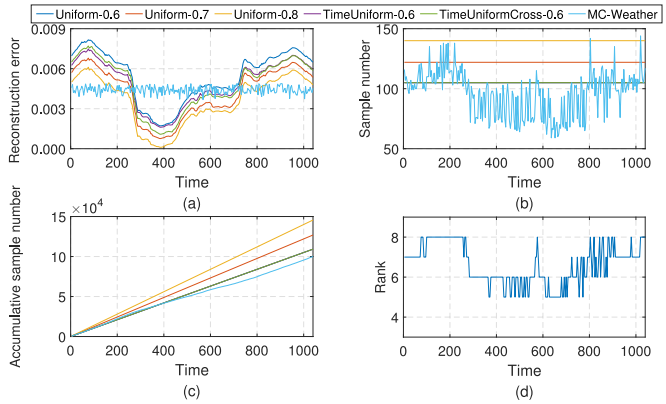


Fig. 18. Performance on trace PM 10.

reconstruction error ratio at low level with the total number of samples lower than the peer sampling strategies.

7 Conclusion

In this paper, we focus on continuous and on-line data gathering in WSNs. Through analyzing datasets of real weather data in ZhuZhou, China, we observe that weather data have the features of low-rank, temporal stability, and relative rank stability. Taking advantage of these structures, we propose an on-line MC-Weather scheme based on matrix completion theory. We prove that the observed relative rank stability is a common feature in continuous data gathering systems. Based on this important feature and our observations, we propose three sample learning principles, which are applied to guide our adaptive sampling algorithm to quickly determine the effective sampling set. To take full advantage of our sample learning principles, we also propose a Uniform Time-slot and Cross Sample model. Compared with the Bernoulli model, we prove that our UTSCS model allows for better data matrix reconstruction. Trace-driven simulations based on real weather data traces and other sensory data traces (PM 2.5 and PM 10) show that MC-Weather can achieve a high accuracy in data recovery with low sensing and communication costs in a dynamic environment.

Acknowledgments

The work is supported in part by the National Natural Science Foundation of China under Grant Nos. 61572184, 61725206, 61472130, and 61472131, in part by the Hunan Provincial Natural Science Foundation of China under Grant No. 2017JJ1010, in part by the US National Science Foundation ECCS 1408247, CNS 1526843, and ECCS 1731238, in part by the Foundation of Key Laboratory of Machine Intelligence and Advanced Computing of the Ministry of Education under Grant No.MSC-201708A, in part by Science and Technology Key Projects of Hunan Province under Grant No.2015TP1004, and in part by the outstanding graduate student innovation fund program of collaborative innovation center of high performance computing.

References

[1] J. Luo, J. Hu, D. Wu, and R. Li, "Opportunistic routing algorithm for relay node selection in wireless sensor networks," *IEEE Trans. Ind. Inform.*, vol. 11, no. 1, pp. 112–121, Feb. 2015.

[2] X. Liu, S. Zhang, and K. Bu, "A locality-based range-free localization algorithm for anisotropic wireless sensor networks," *Telecommun. Syst.*, vol. 62, no. 1, pp. 3–13, 2016.

[3] S. Zhang, X. Liu, J. Wang, J. Cao, and G. Min, "Accurate range-free localization for anisotropic wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 11, no. 3, 2015, Art. no. 51.

[4] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: An acquisitional query processing system for sensor networks," *ACM Trans. Database Syst.*, vol. 30, no. 1, pp. 122–173, 2005.

[5] G. Hua and C. W. Chen, "Correlated data gathering in wireless sensor networks based on distributed source coding," *Int. J. Sensor Netw.*, vol. 4, no. 1, pp. 13–22, 2008.

[6] J. Chou, D. Petrovic, and K. Ramachandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in *Proc. IEEE Soc. Annu. Joint Conf. IEEE Comput. Commun.*, 2003, pp. 1054–1062.

[7] K. Yuen, B. Liang, and L. Baochun, "A distributed framework for correlated data gathering in sensor networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 1, pp. 578–593, Jan. 2008.

[8] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 309–316.

[9] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *Proc. IEEE Soc. Annu. Joint Conf. IEEE Comput. Commun.*, 2003, pp. 1848–1857.

[10] S. Yoon and C. Shahabi, "The clustered aggregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 3, no. 1, 2007, Art. no. 3.

[11] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 7, pp. 1010–1023, Jul. 2007.

[12] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Trans. Sensor Netw.*, vol. 4, no. 1, 2008, Art. no. 4.

[13] X. Xu, X.-Y. Li, P.-J. Wan, and S. Tang, "Efficient scheduling for periodic aggregation queries in multihop sensor networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 690–698, Jun. 2012.

[14] K.-W. Fan, S. Liu, and P. Sinha, "Structure-free data aggregation in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 8, pp. 929–942, Aug. 2007.

[15] K.-W. Fan, S. Liu, and P. Sinha, "Dynamic forwarding over tree-on-DAG for scalable data aggregation in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 10, pp. 1271–1284, Oct. 2008.

[16] H. Luo, Y. Liu, and S. K. Das, "Distributed algorithm for en route aggregation decision in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 1, pp. 1–13, Jan. 2009.

[17] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 134–142.

[18] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 92–101, Mar. 2008.

[19] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proc. Annu. ACM Int. Conf. Mobile Comput. Netw.*, 2009, pp. 145–156.

[20] J. Luo, L. Xiang, and C. Rosenberg, "Does compressed sensing improve the throughput of wireless sensor networks?," in *Proc. IEEE Int. Conf. Commun.*, 2010, pp. 1–6.

[21] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *Proc. Annu. IEEE Commun. Soc. Conf. Senssor Mesh Ad Hoc Commun. Netw.*, 2011, pp. 46–54.

[22] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *Proc. IEEE INFOCOM*, 2012, pp. 603–611.

[23] C. T. Chou, R. Rana, and W. Hu, "Energy efficient information collection in wireless sensor networks using adaptive compressive sensing," in *Proc. IEEE 34th Conf. Local Comput. Netw.*, 2009, pp. 443–450.

[24] K. Xie, et al., "An efficient privacy-preserving compressive data gathering scheme in WSNs," *Inf. Sci.*, vol. 390, pp. 82–94, 2017.

[25] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

[26] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *J. Mach. Learn. Res.*, vol. 99, pp. 2057–2078, 2010.

[27] A. Eriksson and A. Van Den Hengel, "Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l_1 norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 771–778.

[28] X. Deng, T. He, L. He, J. Gui, and Q. Peng, "Performance analysis for IEEE 802.11s wireless mesh network in smart grid," *Wireless Pers. Commun.*, vol. 90, no. 4, pp. 1–19, 2017.

[29] X. Deng, L. He, X. Li, Q. Liu, L. Cai, and Z. Chen, "A reliable QoS-aware routing scheme for neighbor area network in smart grid," *Peer-to-Peer Netw. Appl.*, vol. 9, no. 4, pp. 616–627, 2016.

[30] J. Luo, Y. Guo, S. Fu, K. Li, and W. He, "Virtual resource allocation based on link interference in Cayley wireless data centers," *IEEE Trans. Comput.*, vol. 64, no. 10, pp. 3016–3021, Oct. 2015.

[31] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1417–1429, May 2017.

[32] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[33] Y. Li, K. Xie, and X. Wang, "Pushing towards the limit of sampling rate: Adaptive chasing sampling," in *Proc. IEEE 12th Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2015, pp. 398–406.

[34] K. Xie, et al., "Decentralized context sharing in vehicular delay tolerant networks with compressive sensing," in *Proc. IEEE 36th Int. Conf. Distrib. Comput. Syst.*, 2016, pp. 169–178.

[35] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Signals Syst. Comput.*, 1993, pp. 40–44.

[36] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012.

[37] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 310–316, Apr. 2010.

[38] B. Zhang, X. Cheng, N. Zhang, Y. Cui, Y. Li, and Q. Liang, "Sparse target counting and localization in sensor networks based on compressive sensing," in *Proc. IEEE INFOCOM*, 2011, pp. 2255–2263.

[39] K. Xie, X. Wang, J. Wen, and J. Cao, "Cooperative routing with relay assignment in multiradio multihop wireless networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 859–872, Apr. 2016.

[40] K. Xie, X. Wang, X. Liu, J. Wen, and J. Cao, "Interference-aware cooperative communication in multi-radio multi-channel wireless networks," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1528–1542, May 2016.

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

[42] E. J. Candes and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[43] J. Cheng, Q. Ye, H. Jiang, D. Wang, and C. Wang, "STCDG: An efficient data gathering algorithm based on matrix completion for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 2, pp. 850–861, Feb. 2013.

[44] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 836–844.

[45] C. Qiu, N. Vaswani, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 5954–5958.

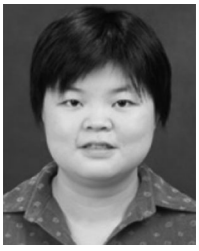
[46] G. Xie, K. Xie, J. Huang, X. Wang, Y. Chen, and J. Wen, "Fast low-rank matrix approximation with locality sensitive hashing for quick anomaly detection," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

[47] K. Xie, C. Peng, X. Wang, G. Xie, and J. Wen, "Accurate recovery of internet traffic data under dynamic measurements," in *Proc. IEEE INFOCOM*, 2017, pp. 1–9.

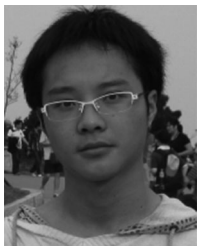
[48] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[49] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 763–770.

[50] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, "Robust photometric stereo via low-rank matrix completion and recovery," in *Proc. Asian Conf. Comput. Vis.*, 2011, pp. 703–717.
[51] G. Gürsun and M. Crovella, "On traffic matrix completion in the internet," in *Proc. Conf. Internet Meas. Conf.*, 2012, pp. 399–412.
[52] K. Li, Q. Dai, W. Xu, J. Yang, and J. Jiang, "Three-dimensional motion estimation via matrix completion," *IEEE Trans. Syst. Man Cybern. Part B: Cybern.*, vol. 42, no. 2, pp. 539–551, Apr. 2012.
[53] Z. Weng and X. Wang, "Low-rank matrix completion for array signal processing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2012, pp. 2697–2700.
[54] R. Ma, N. Barzigar, A. Roozgard, and S. Cheng, "Decomposition approach for low-rank matrix completion and its applications," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1671–1683, 2014.
[55] Y. Liao, W. Du, P. Geurts, and G. Leduc, "DMFSGD: A decentralized matrix factorization algorithm for network distance prediction," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1511–1524, Oct. 2013.
[56] K. Xie, L. Wang, X. Wang, J. Wen, and G. Xie, "Learning from the past: Intelligent on-line weather monitoring based on matrix completion," in *Proc. IEEE 34th Int. Conf. Distrib. Comput. Syst.*, 2014, pp. 176–185.
[57] K. Xie, et al., "Recover corrupted data in sensor networks: A matrix completion solution," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1434–1448, May 2017.
[58] K. Xie, et al., , "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2443–2451.
[59] H. Zhou, D. Zhang, and K. Xie, "Accurate traffic matrix completion based on multi-gaussian models," *Comput. Commun.*, vol. 102, pp. 165–176, 2017.
[60] Z. Huibin, Z. Dafang, X. Kun, and W. Xiaoyang, "Data reconstruction in internet traffic matrix," *China Commun.*, vol. 11, no. 7, pp. 1–12, 2014.
[61] M. Fornasier, H. Rauhut, and R. Ward, "Low-rank matrix recovery via iteratively reweighted least squares minimization," *SIAM J. Optimization*, vol. 21, no. 4, pp. 1614–1640, 2011.
[62] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.
[63] D. Goldfarb and S. Ma, "Convergence of fixed-point continuation algorithms for matrix rank minimization," *Found. Comput. Math.*, vol. 11, no. 2, pp. 183–210, 2011.
[64] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Math. Program.*, vol. 128, no. 1/2, pp. 321–353, 2011.
[65] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
[66] I. Markovsky, *Low Rank Approximation: Algorithms, Implementation, Applications*. Berlin, Germany: Springer, 2011.
[67] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.
[68] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.
[69] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When Urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 1436–1444.
[70] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Comput. Netw.*, vol. 45, no. 3, pp. 245–259, 2004.



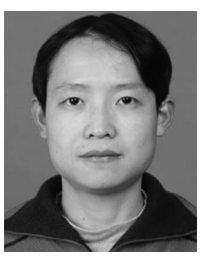
Kun Xie received the PhD degree in computer application from Hunan University, Changsha, China, in 2007. She worked as a postdoctoral fellow in the Department of Computing, Hong Kong Polytechnic University from Dec. 2007 to Feb. 2010. She worked as a visiting researcher in the Department of Electrical and Computer Engineering, State University of New York at Stony Brook from Sep. 2012 to Sep. 2013. She is currently a professor with Hunan University, Changsha, China. Her research interests include wireless network and mobile computing, network management and control, cloud computing and mobile cloud, and big data.



Lele Wang is working toward the PhD degree at Hunan University, Changsha, China.

Xin Wang received the PhD degree in electrical and computer engineering from Columbia University, New York, New York. She is currently an associate professor in the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, New York. Before joining Stony Brook, she was a member of technical staff in the area of mobile and wireless networking in the Bell Labs Research, Lucent Technologies, New Jersey, and an assistant professor in the Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, New York. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, as well as networked sensing and detection. She has served on the executive committee and technical committee of numerous conferences and funding review panels, and served as the associate editor of the *IEEE Transactions on Mobile Computing*. She achieved the NSF Career Award in 2005, and ONR Challenge Award in 2010. She is a member of the IEEE.

Gaogang Xie received the BS degree in physics, and the MS and PhD degrees in computer science all from Hunan University, in 1996, 1999, and 2002, respectively. He is currently a professor and director of the Network Technology Research Center with the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China. His research interests include Internet architecture, packet processing and forwarding, and Internet measurement.



Jigang Wen received the PhD degrees in computer application from Hunan University, China, in 2011. He worked as a research assistant in the Department of Computing, Hong Kong Polytechnic University from 2008 to 2010, respectively. He is now a postdoctoral fellow in the Institute of Computing Technology, Chinese Academy of Science, China. His research interests include wireless network and mobile computing, and high speed network measurement and management.