Learning from the Past: Intelligent On-Line Weather Monitoring based on Matrix Completion

Kun Xie [1,2,3], Lele Wang[1], Xin Wang[3], Jigang Wen[4], Gaogang Xie[4]
[1] College of Computer Science and Electronics Engineering, Hunan University, Changsha, China
[2] State key Laboratory of Networking and Switching Technology, Beijing Univ. of Posts and Telecomm., China
[3] Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA
[4] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
xiekun@hnu.edu.cn, wanglele2012@hnu.edu.cn, xwang@ece.sunysb.edu, wenjigang@ict.ac.cn, xie@ict.ac.cn

***Abstract*—Matrix completion has emerged very recently and provides a new venue for low cost data gathering in WSNs. Existing schemes often assume that the data matrix has a known and fixed low-rank, which is unlikely to hold in a practical monitoring system such as weather data gathering. Weather data varies in temporal and spatial domain with time. By analyzing a large set of weather data collected from 196 sensors in Zhu Zhou, 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 reduce the cost for sensing, communication and computation. We perform extensive simulations based on the real weather data sets and the simulation results validate the efficiency and efficacy of the proposed scheme.**

***Index Terms*—matrix completion; wireless sensor network; data gathering;**

I. Introduction

Continuous weather data gathering is important for weather reporting and a typical application of wireless sensor networks (WSN). In the traditional data gathering approach [1], each sensor node senses and sends data to a sink periodically, 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 weather gathering 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 [2]–[4], in-network collaborative wavelet transform [5], [6] and clustered data aggregation [7]–[10]. 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 [11]–[17]. 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 rather than just events for system management purpose.

With the rapid progress of sparse representation, matrix completion [18]–[20], 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. Matrix completion brings the benefits of small set of samples at sensor nodes without introducing excessive computational and traffic overheads, which meets the limited resource constraint in WSN. 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 assumption is unlikely to hold for real weather data gathering due to the dynamics of weather data, and our observation on data trace indicates that rank varies in temporal or spatial domain with time.

To study the feature of weather data, we have deployed 196 sensors in Zhu Zhou, China, to collect weather data for more than two years. From large real 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; otherwise, the reconstruction may fail. 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, which 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 reconstruction accuracy. 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:

- Based on the trace analysis of the large set of real weather data collected, we 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 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 sampling set to apply with 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 feature for higher matrix completion performance.
- Through comprehensive simulations with real data traces, we show that our MC-Weather scheme can accurately acquire weather data with 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. The techniques proposed in this paper can be applied to other monitoring systems. The proposed MC-Weather scheme does not depend on the choice of underlying matrix reconstruction algorithms.

The rest of this paper is organized as follows. We introduce the related work in Section II. The fundamentals of matrix completion and problem formulation are presented in Section III. We present our empirical study with real weather data in Section IV. The proposed MC-Weather is presented in Section V. Finally, we evaluate the performance of the proposed MC-Weather through extensive simulations in Section VI, and conclude the work in Section VII.

II. RELATED WORK

Structure and redundancy in data are often synonymous with sparsity. There exist two typical sparsity representation techniques, compressive sensing and matrix completion.

Compressive Sensing (CS) is a technique that can accurately recover a vector from a subset of samples given that the vector is sparse [12], [21] 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. Compressive sensing has two features, universal sampling and decentralized simple encoding, which makes it a new paradigm for data gathering in sensor networks [11]–[17].

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. However, some matrices have some inherent structure (i.e. the weather matrix in this paper), low cost data gathering in WSN has lots of space to improve. In addition, some applications do not have clear sparse features, and we may often need to get more complete data rather than just detect some sparse events.

On the heels of compressed sensing, matrix completion has emerged very recently [18]–[20], [22]. Candès et al. [18] 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 [22]. Matrix completion brings new opportunities to fully exploit the low-rank property in various associated applications [23]–[30].

Existing schemes based on matrix completion are mostly designed for off-line execution and can not apply in on-line weather data gathering with dynamic environment changes. 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 in a practical system with dynamic environment and rank variations.

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 slots 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 weather monitoring.

III. PRELIMINARY AND PROBLEM FORMULATION

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

A. 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 [18]–[20], [22]. 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. The subset Ω is formed with randomly selected entries of the matrix, and the sampling operator $P_\Omega : R^{n_1 \times n_2} \to R^{n_1 \times n_2}$ 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 [18]

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

where rank(.) denotes the rank of a matrix, X is the variable matrix.

However, solving this rank minimization problem in (2) is often impractical because it is NP-hard. Then [18] proves that most matrices M of rank r can be perfectly recovered by solving the optimization problem

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

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

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

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). our proposed MC-Weather scheme does not depend on the underlying reconstruction approach. We choose the singular value thresholding (SVT) approach [31] to reconstruct the matrix.

B. 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, which contains the data 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 measurement data in the next section. MC-Weather measures the weather condition only at a subset of the locations in a given time slot and vary the data collection locations in different time slots. Rather than randomly select the measurement locations as instructed by conventional matrix completion theory, we find that the performance can be improved if we could select the collection points 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

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

Accordingly, a Binary Sampling Matrix $B_{N\times T}(t)$ can be defined as $B_{N\times T}(t) = \left[\vec{B}(t), \vec{B}(t+1), \cdots, \vec{B}(t+T-1)\right]$, 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 III-A, 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

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

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

Although the literature work on matrix completion provide some solutions to recovering data with a limited number of samples, existing schemes mostly assume the rank of the sensory matrix is low and has a constant rank 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 matrix completion theory in the 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 weather monitoring data to better understand the structure and characteristics of weather data in the next section.

IV. 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. Fig.2 shows the deployed sensor node. 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 Zhu Zhou. Each data element includes weather data of rain, temperature, and wind. Specially, we choose rain data to analyze because Zhu Zhou 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.

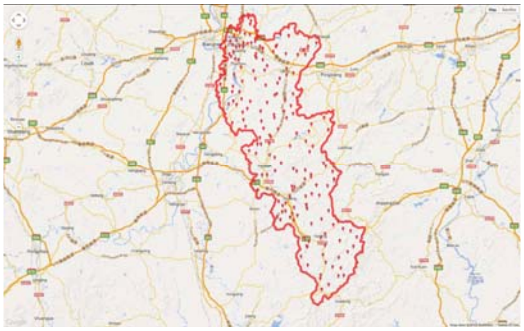

Fig. 1. Weather sensor deployment in Zhu Zhou, China

Fig. 2. Sensor node deployed in Zhu Zhou, China

A. *Low-rank feature*

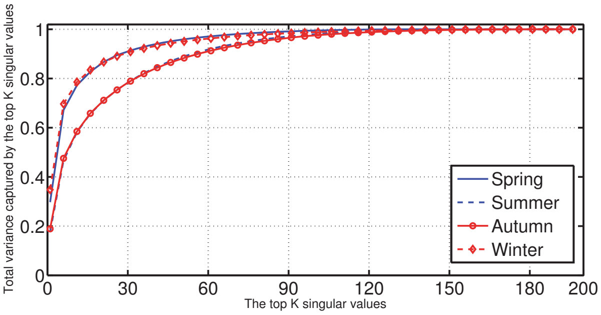

Fig. 3. Fraction captured by top k singular values

Weather data collected over different locations and time slots are not independent. There exists inherent data redundancy. We first apply 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 is an $N \times N$ unitary matrix, V is a $T \times T$ unitary matrix, and Σ is a $N \times T$ diagonal matrix with the diagonal elements (i.e. the singular values) organized in the decreasing order (i.e. $\Sigma = diag\left(\sigma_1, \sigma_2, \cdots, \sigma_r, 0, \cdots, 0\right)$). The rank of a matrix X, denoted by r, is equal to the number of its non-zero singular values. A matrix is low-rank if its $r \ll \min\{N, T\}$.

In Eq(8), the singular value σ_i also indicates the energy of the i-th principal component. According to PCA (Principal components analysis), if a matrix has low-rank, its top k singular values occupy the total or near-total energy $\sum\limits_{i=1}^{k} \sigma_i^2 \approx \sum\limits_{i=1}^{r} \sigma_i^2$. The metric we use is the fraction of the total variance captured by the top k singular values:

$$g\left(k\right) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{r} \sigma_i^2 \tag{9}$$

Fig.3 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% 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. The low-rank feature is the prerequisite for using matrix completion.

B. *Temporal stability*

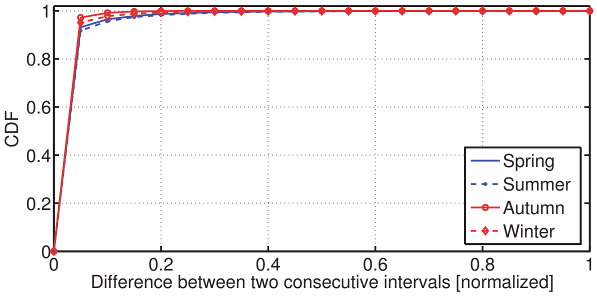

Fig. 4. Temporal stability feature

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

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

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

$$\Delta gap(i, j) = \frac{|x_{ij} - x_{i,j-1}|}{\max\limits_{1\leqslant i\leqslant N, 2\leqslant j\leqslant T} |x_{ij} - x_{i,j-1}|} \tag{11}$$

where $\max\limits_{1\leqslant i\leqslant N, 2\leqslant j\leqslant T} |x_{ij} - x_{i,j-1}|$ 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.4. 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% $\Delta gap(i,j)$ are very small (< 0.05). These results indicate that temporal stability exists in real environments. In Section V-D2, we design our cross sample model by utilizing this feature.

C. Rank-stability

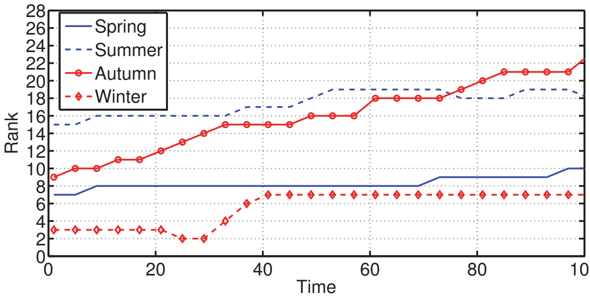

Fig. 5. Rank feature of weather data

We plot the rank of the consecutive weather matrix in Fig.5 by varying the starting time slot from 0 to 100 to further investigate the rank feature. 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 V-B, we will exploit the relative rank stability in our learning algorithm for more efficient on-line weather gathering.

V. 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 and long lifespan of sensors, and reduced data transmissions in the network.

A. Rank of adjacent matrices

To support continuous weather gathering and reduce the computation cost for reconstructing the weather matrix, our 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 which contains the current time slot the *active measurement window*.

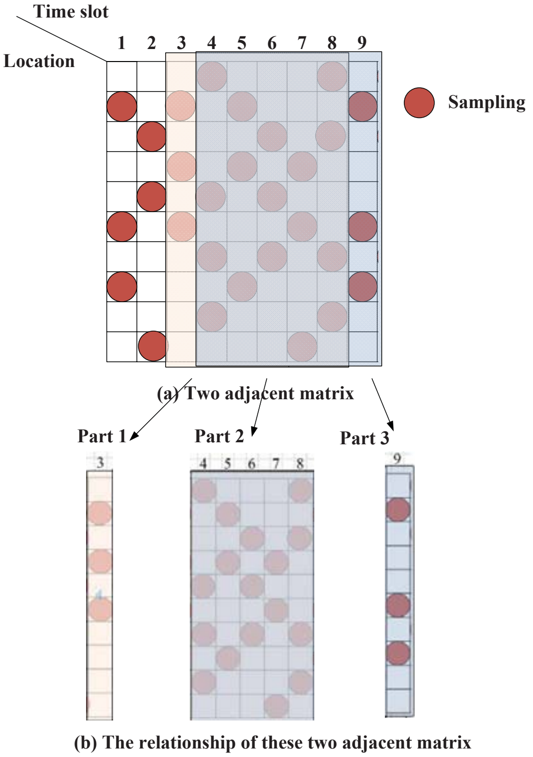

Fig. 6. Slide window based weather gathering

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

From the matrix completion theory, the rank of the matrix has 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 window because the rank of an active measurement window is unknown.

As shown in Fig.6(b), obviously, most columns of these two matrices are the same except one column. Therefore, there exists a strong relationship between these two matrices. Before we discuss the relationship of these two adjacent matrices 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: Due to the limited space, the proof is omitted.

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

$$r - 1 \leqslant rank(X_{N\times T}(t+1)) \leqslant r + 1. \tag{14}$$

Proof: Due to the limited space, the proof is omitted.

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

B. Sample learning principle

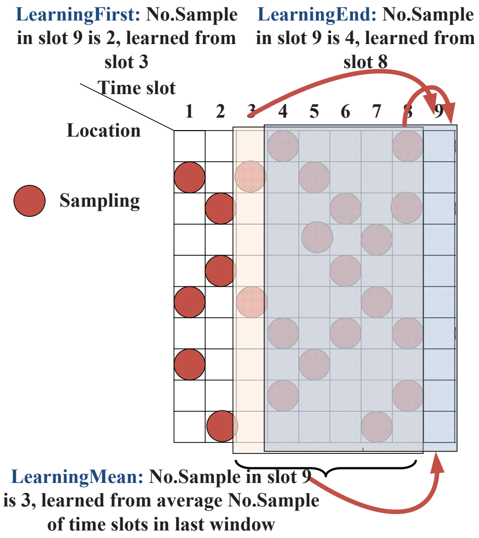
LearningFirst: No.Sample in slot 9 is 2, learned from slot 3
LearningEnd: No.Sample in slot 9 is 4, learned from slot 8
Time slot
Location
1
2
3
4
5
6
7
8
9
Sampling
LearningMean: No.Sample in slot 9 is 3, learned from average No.Sample of time slots in last window

Fig. 7. Different 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.7, 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.7, the initial sampling number in slot 9 is learned 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.7, 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.7, 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 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, it is also sufficient to set the number of samples in slot 9 to 2, the same to 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.

C. Adaptive sampling

We propose our adaptive sampling algorithm based on the matrix completion and learning principles below.

Algorithm 1 The matrix completion based adaptive sampling algorithm

```
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 V-D2, select sampling locations and initialize $\vec{B}(t)$ with $\left|\vec{B}(t)\right| = C$. The sink announces the sampling schedule according to $\vec{B}(t)$.
Once the 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 $\varepsilon$ as
    $\varepsilon = \frac{\sqrt{\sum_{i,j,B_{ij}(t)=1}\left(M_{ij}(t) - \hat{X}_{ij}(t)\right)^2}}{\sqrt{\sum_{i,j,B_{ij}(t)=1} M_{ij}(t)^2}}$    (15)
while $|\varepsilon - \varepsilon_b| > \beta$ do
    if $\varepsilon - \varepsilon_b > 0$ then
        Add $\alpha C(\varepsilon - \varepsilon_b)$ extra measurements according to cross-based sampling principle in Section V-D2, and update $\vec{B}(t)$ and $C = C + \alpha C(\varepsilon - \varepsilon_b)$.
    else
        Delete $\alpha C(\varepsilon_b - \varepsilon)$ out of measurements at the sink to look for the appropriate number of samples to take in future time slots. and update $\vec{B}(t)$ and $C = C - \alpha C(\varepsilon_b - \varepsilon)$.
    end if
    Based on the updated $\vec{B}(t)$, calculate the reconstruction error $\varepsilon$ according to Eq(15).
end while
The sink stores $\vec{B}(t)$ to indicate the effective sampling in time slot $t$.
```

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 $\hat{X}_{N\times T}\left(t-T+1\right)$ and calculate the reconstruction error ε according to Eq(15).

If the error is low so the reconstruction can reach the accuracy requirement, we consider $\hat{X}$ as a successful recovery and our algorithm goes to step 7 to reduce the extra measurements and find the number of samples needed in future time slots; otherwise, our algorithm goes to the step 5 to determine the number of supplemental measurements to take.

The large recovery error can be due to the variation of environmental conditions thus the rank change of the matrix. In our MC-weather system, each time slot is one hour. If the sink finds sample scheduled in the current time slot is not sufficient to reconstruct the matrix, it can instruct sensors to take additional samples. 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 error bound ε_b. We add $\alpha C(\varepsilon-\varepsilon_b)$ extra measurements according to cross-based sampling principle to present in Section V-D2, and update $C=C+\alpha C(\varepsilon-\varepsilon_b)$ until the error gap is smaller than β. The larger the error gap is, the higher probability the rank increases. Additional measurements are needed to capture the rank variation to more accurately reconstruct the matrix.

In step 7, if the recovery error ε is below the error bound ε_b, it indicates that the current reconstruction with C samples already satisfies the accuracy requirement. When the rank of the weather matrix in an active measurement window decreases, the number of samples to take can also be reduced. Similar to step 6, we propose to adapt the sampling number according to the recovery error gap. Among C measurements, we delete $\alpha C(\varepsilon_b-\varepsilon)$ out of measurements at sink to look for the number of samples needed in future slots and update $C=C-\alpha C(\varepsilon_b-\varepsilon)$ until the error gap 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. From step 3 to step 10, our adaptive sampling algorithm attempts to identify the effective number of samples to take in future time slots based on the rank of the data matrix in the active measurement window, which will guide the future weather monitoring based on the learning principles proposed in Section V-B.

D. 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 history measurements to guide the sampling process. We introduce a training phase in Section V-D1 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 V-D2.

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 $X_{N\times T}\left(1\right)$ and the rank of $r=rank\left(X_{N\times T}\left(1\right)\right)$. 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 [32], 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 $p=m/\left(n_1\times n_2\right)$ (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 $\left\lceil \frac{m}{T}\right\rceil$.

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

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.8 shows an example of our Uniform Time-slot and cross-based sampling model. The training measurement window is $X_{10\times10}\left(1\right)$ and the required sampling number is

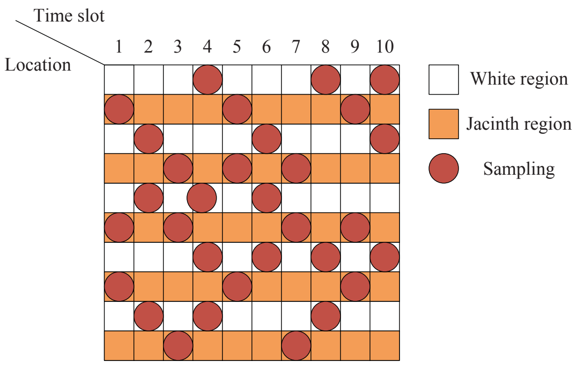

Fig. 8. UTSCS Sampling model.

$m = 30$. According to the model, each time slot should have 30/10=3 effective samples. Moreover, to implement uniform sampling by avoiding sampling the same location in adjacent time slots, the locations are divided into two parts, white part and jacinth part. In time slot 1, effective samples have high priority to take in jacinth part, while in time slot 2, effective samples have high priority to take in white part.

3) Sample model analysis

The key difference between our Uniform Time-slot and cross based sampling model (UTSCS) and the other two models 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. 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 Theorem 3, we will show that the probability of missing an entire row under our UTSCS model is smaller than that under the Bernoulli model.

Let F be the event that an entire row is missed to sample. Under the Bernoulli model, as each sample is taken independently, the probability of event F is $P_{Bernoulli}\left(F\right) = \left(1 - \frac{m}{N \times T}\right)^{T} = \left(\frac{N \times T - m}{N \times T}\right)^{T}$, where $\frac{m}{N \times T}$ is the probability of each entry in the matrix is sampled.

Under the UTSCS Sampling model, the probability of event F is $P_{UTSCS}\left(F\right) = \left(\frac{C_{m/T}^{N/2-1}}{C_{m/T}^{N/2}}\right)^{T/2} = \left[\left(\frac{N \times T - 2m}{N \times T}\right)^{1/2}\right]^{T}$.

Theorem 3: When $N \geqslant \frac{2}{\ln 10} T$, the ratio of probability $\frac{P_{Bernoulli}(F)}{P_{UTSCS}(F)} = \frac{\left(\frac{N \times T - m}{N \times T}\right)^{T}}{\left[\left(\frac{N \times T - 2m}{N \times T}\right)^{1/2}\right]^{T}}$ satisfies

$$1 < \frac{P_{Bernoulli}\left(F\right)}{P_{UTSCS}\left(F\right)} < e^{1/2} \qquad (16)$$

Proof: Due to the limited space, the proof is omitted.

From [32], we know that sampling according to Bernoulli model has been analyzed and shown to be able to recover the matrix satisfactorily. From the Theorem 3, we can conclude that compared with Bernoulli model, our UTSCS sample model has better performance for matrix completion.

E. Complete MC-weather gathering scheme

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. Then the Uniform Time-slot and cross Sampling model is applied to identify the effective sample sets within the training window.

(2)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. It then adapts the sampling set following the adaptive algorithm in Section V-C to accurately reconstruct the weather matrix in the presence of the change of environmental conditions and accordingly the rank of the weather data matrix.

VI. Performance Evaluations

In this section, we first introduce the methodology and simulation setup, and then analyze our performance results.

A. Methodology and experimental setup

To evaluate the performance of our MC-Weather scheme, we have performed extensive simulations driven by real weather traces collected by our deployed 196 sensors. Specifically, we chose the rain traces gathered from July 1 to August 31st, 2012.

We implement four weather gathering schemes in our simulations. The first scheme is our MC-Weather scheme in which the effective samples in the training windows are obtained according to our UTSCS Sampling model proposed in Section V-D and the effective samples in each new time slot is set according to Algorithm 1. Especially, according to [18], to well control the reconstruction error of matrix completion, the low bound error and the error gap in Algorithm 1 are set to ε_b= 0.4%, β= 0.05%. 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 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, uniform-time slot sampling model proposed in Section V-D1 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 V-D2 is applied to identify the sample set in new time slot, denoted as TimeUniformCross-0.6.

B. Simulation results

1) Estimation error

In Fig.9, 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 varies with time, and sampling with a fixed ratio is not suitable for dynamic weather data gathering. Even for Uniform-0.8 (sampling with the largest ratio), the error rate around time slot 1300 raises up to three times higher, which is much larger than the error bound. In contrast, the error rate in MC-weather can be well controlled to be around the error bound, which demonstrates that MC-weather can successfully adapt the sampling rate in response to 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 indicates that the sampling model taken by TimeUniform-0.6 is better to apply with 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 helps achieve better data gathering performance.

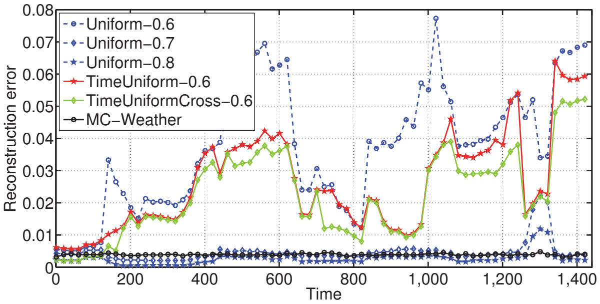

Fig. 9. Reconstruction error.

2) Sample number

Fig.10 and Fig.11 compare the number of samples and the accumulative number of samples taken under different schemes. In consistence with the results shown in Fig.9, the curves in all the schemes in Fig.10 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 the rank variation to accurately recover data matrix while reducing the sampling overhead. We observe higher sampling number change around time 1300.

Fig.11 also demonstrates that our uniform Time slot and cross sample model is a good for matrix completion. 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.9).

3) Impact of sampling learning principles

To evaluate the performance with different sample learning principles proposed in Section V-C, 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

$$Gap_{ratio}\left[i\right] = \frac{\left|C_{Learning}\left[i\right] - C_{Effective}\left[i\right]\right|}{C_{Effective}\left[i\right]} \tag{17}$$

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

We plot the CDF of $Gap_{ratio}\left[i\right]$ in Fig.12. The X-axis presents the gap ratio. The Y-axis presents the cumulative probability. For the LearningEnd principle, $> 90\%$ probability the values of $Gap_{ratio}\left[i\right]$ 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.

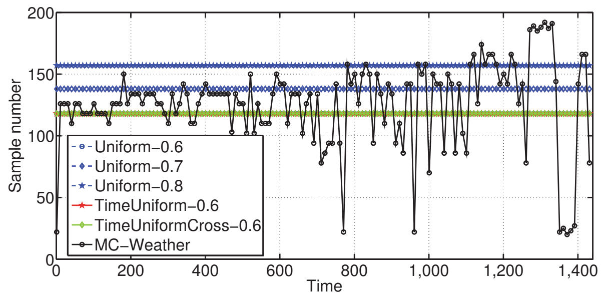

Fig. 10. Adaptive sample number.

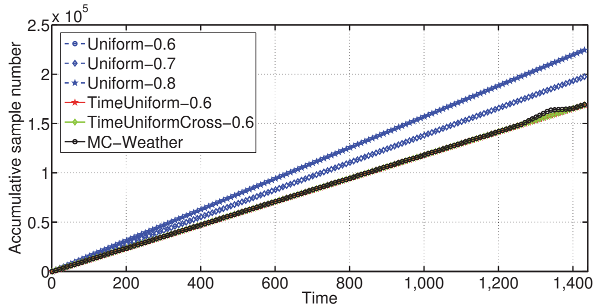

Fig. 11. Total sample number.

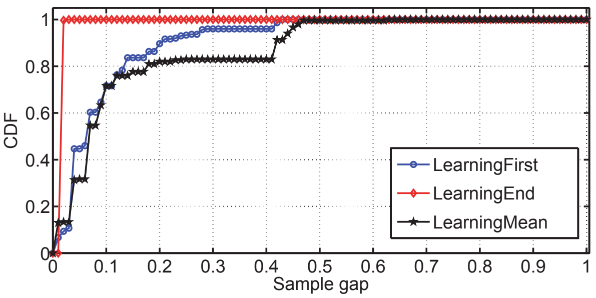

Fig. 12. Sample gap under different sample learning principle.

VII. Conclusion

In this paper, we focus on continuous and on-line data gathering in WSNs. Through analyzing datasets of real weather data in Zhu Zhou, China, we observe that weather data have the features of low-rank, temporal stability, and relative rank stability. By 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 common feature in continuous data gathering systems. Based on this important feature and our observations, we propose three sample learning principles, based on which we design our adaptive sampling algorithm to quickly determine the effective sampling set. To take the full advantage of our sample learning principles, we also propose a Uniform Time-slot and Cross Sample model (UTSCS). 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 show that MC-Weather successfully achieves required high accuracy in data recovery with low sensing and communication costs in a dynamic environment.

Acknowledgment

The work is supported by the open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under Grant No. SKLNST-2013-1-04, the National Natural Science Foundation of China under Grant Nos. 61133015 and 61173167, U.S. National Science Foundation under Grant Nos. ECCS-1231800 and CNS 1247924, and the National High-Tech Research and Development Plan of China under Grant No.2012AA01A301-01.

References

[1] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "Tinydb: An acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems (TODS)*, vol. 30, no. 1, pp. 122–173, 2005.

[2] G. Hua and C. W. Chen, "Correlated data gathering in wireless sensor networks based on distributed source coding," *International Journal of Sensor Networks*, vol. 4, no. 1, pp. 13–22, 2008.

[3] J. Chou, D. Petrovic, and K. Ramachandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in *INFOCOM 2003*.

[4] K. Yuen, B. Liang, and L. Baochun, "A distributed framework for correlated data gathering in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 578–593, 2008.

[5] 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 *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006.

[6] M. Crovella and E. Kolaczyk, "Graph wavelets for spatial traffic analysis," in *INFOCOM 2003*.

[7] S. Yoon and C. Shahabi, "The clustered aggregation (cag) technique leveraging spatial and temporal correlations in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 3, no. 1, p. 3, 2007.

[8] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 1010–1023, 2007.

[9] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 1, p. 4, 2008.

[10] X. Xu, X.-Y. Li, P.-J. Wan, and S. Tang, "Efficient scheduling for periodic aggregation queries in multihop sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 3, pp. 690–698, 2012.

[11] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, "Compressive wireless sensing," in *Proceedings of the 5th international conference on Information processing in sensor networks*. ACM, 2006.

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

[13] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009.

[14] J. Luo, L. Xiang, and C. Rosenberg, "Does compressed sensing improve the throughput of wireless sensor networks?" in *ICC 2010*. IEEE.

[15] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *SECON 2011*. IEEE.

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

[17] C. T. Chou, R. Rana, and W. Hu, "Energy efficient information collection in wireless sensor networks using adaptive compressive sensing," in *LCN 2009*. IEEE.

[18] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[19] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *The Journal of Machine Learning Research*, vol. 99, pp. 2057–2078, 2010.

[20] 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 *CVPR 2010*. IEEE, 2010.

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

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

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," DTIC Document, Tech. Rep., 2000.

[24] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[25] 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 *CVPR 2010*. IEEE.

[26] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma, "Robust photometric stereo via low-rank matrix completion and recovery," in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 703–717.

[27] 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 Transactions on Wireless Communications*, vol. 12, no. 2, pp. 850–861, 2013.

[28] G. Gürsun and M. Crovella, "On traffic matrix completion in the internet," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 399–412.

[29] K. Li, Q. Dai, W. Xu, J. Yang, and J. Jiang, "Three-dimensional motion estimation via matrix completion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 539–551, 2012.

[30] Z. Weng and X. Wang, "Low-rank matrix completion for array signal processing," in *ICASSP 2012*. IEEE.

[31] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[32] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.