# Interpolating the Missing Values for Multi-Dimensional Spatial-Temporal Sensor Data: A Tensor SVD Approach

Peipei Xu
School of Electronic Engineering
UESTC
Chengdu, China 611731
peipei.xu6@gmail.com

Wenjie Ruan
Department of Computer Science
University of Oxford
Oxford, UK OX1 3QD
wenjie.ruan@cs.ox.ac.uk

Quan Z. Sheng
Department of Computing
Macquarie University
Sydney, Australia NSW 2109
michael.sheng@mq.edu.au

Tao Gu
School of CS & IT
RMIT University
Melbourne, Australia VIC 3000
tao.gu@rmit.edu.au

Lina Yao
School of Computer Science and Engineering, UNSW
Sydney, Australia NSW 2052
lina.yao@unsw.edu.au

## ABSTRACT

With the booming of the Internet of Things, enormous number of smart devices/sensors have been deployed in the physical world to monitor our surroundings. Usually those devices generate high-dimensional geo-tagged time-series data. However, these sensor readings are easily missing due to the hardware malfunction, connection errors or data corruption, which severely compromise the back-end data analysis. To solve this problem, in this paper we exploit tensor-based Singular Value Decomposition method to recover the missing sensor readings. The main novelty of this paper lies in that, *i)* our tensor-based recovery method can well capture the multi-dimensional spatial and temporal features by transforming the irregularly deployed sensors into a sensor-array and folding the periodic temporal patterns into multiple time dimensions, *ii)* it only requires to tune one key parameter in an unsupervised manner, and *iii)* Tensor Singular Value Decomposition structure is more efficient on representation of high-dimension sensor data than other tensor recovery methods based on tensor's vectorization or flattening. The experimental results in several real-world one-year air quality and meteorology datasets demonstrate the effectiveness and accuracy of our approach.

## CCS CONCEPTS

**•Hardware → Signal processing systems; Sensor applications and deployments;**

## KEYWORDS

Sensor Data Recovery, Tensor Completion, t-SVD, ADMM

**ACM Reference format:**
Peipei Xu, Wenjie Ruan, Quan Z. Sheng, Tao Gu, and Lina Yao. 2017. Interpolating the Missing Values for Multi-Dimensional Spatial-Temporal Sensor Data: A Tensor SVD Approach. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, Melbourne, VIC, Australia, November 7–10, 2017 (MobiQuitous 2017),* 10 pages.
DOI: 10.1145/3144457.3144474


*MobiQuitous 2017, Melbourne, VIC, Australia*

DOI: 10.1145/3144457.3144474

## 1 INTRODUCTION

In the era of the Internet of Things (IoT), massive number of smart devices and sensors have been installed in our surrounding environments. According to a report, there are more than 1.9 billion sensory devices launched into the physical world each week and there will be rapidly increased into 9 billion by 2018. Such tremendous number of smart devices enable us to monitor, analyze and understand our physical world, and ultimately support us to better manage various facets in the society [9]. For example, with fine-particles (PM 2.5) data across a city, we can understand the trend of environmental pollution and then make a better plan to manage the factories and transportation so that we can reduce the air pollution. Given large-scale trajectory data from the GPS in taxicab and smart bicycles, we can predict the crowd flows in a city so that we can prevent the dangerous stampede by traffic control and warning people in advance. One of the important prerequisites for enabling those promising applications is that we can accurately and continuously access sensory data from the environments.

However, in practice, those sensory data usually suffer from reading-missing or value-lost due to unexpected hardware failures, communication conflicts or harsh environments *etc.*. This disturbing phenomenon not only decreases the real-time monitoring capability of the sensory devices but also further compromises the accuracy of back-end data analysis. Therefore, how to accurately yet efficiently recover the missing sensor data deserves our careful exploration. However, recovering missing values for a high-dimensional geo-tagged time-series data is a challenging task. Firstly, the sensor readings are normally absent randomly which may be missing at consecutive timestamps, or lost at a certain time-stamp for the whole geographic area. This disturbing situation makes the traditional regression-based methods or non-negative matrix decomposition method useless due to, for example, one or many columns and rows are missing at the same time. Secondly, in practice, those

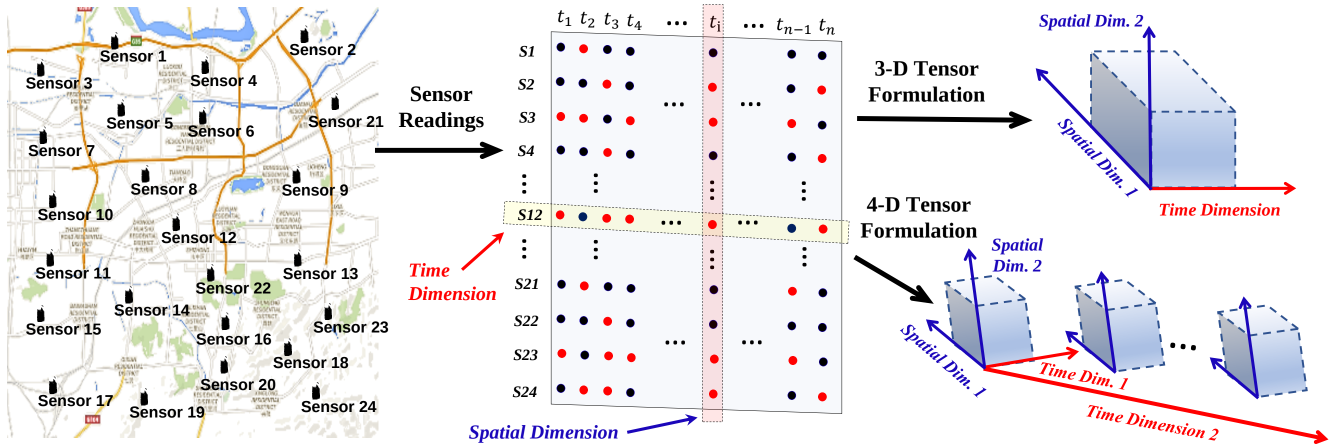



**Figure 1: The idea of tensor formulation:** ***Many tensors are irregularly deployed throughout the city (shown by the first figure), and they generate huge amount of time series data that normally have two dimensions - time and spatial dimensions (shown by the second figure). Those sensor readings are easily missing or lost (represented by the red dots), so this paper aims to recover those missing sensor readings. The idea is to formulate the data as a 3-order tensor such as two spatial dimensions (*****i.e.,** ***longitude and latitude) plus one time dimension, or 4-order tensor such as two spatial dimensions and two time dimensions (*****e.g.,** ***hours*** $\times$ ***days).***

sensor data are generated by sensors deployed in different locations (*e.g.,*with different latitudes and longitudes, even altitudes) so that they normally exhibit significant non-nonlinearities which not only strongly relate to the time dimension but also highly depend on their spatial attributes (*i.e.,* latitudes, longitudes or altitudes).

To deal with aforementioned challenges, many methods for recovering missing sensor readings are proposed. The most widely adopted solutions are based on filtering algorithms such as Median Filtering, Kriging, Kalman Filtering [12], or built upon regression methods with various complexities including ARIMA (AutoRegressive Integrated Moving Average), SVR (Support Vector Regression) [30], kNN (k-Nearest Neighbors) [38] *etc.*. Those methods, however, can only learn spatial or temporal attribute, and are insufficient to capture data's global dependencies due to the limitation of their model structures (only quantifying the local or regional data points in terms of time or spatial attributes). Another popular technique is to borrow the idea from recommendation that formulates the multi-dimensional sensor readings as a matrix (*e.g.,* column represents sampling times and row indicates different locations) and then utilizes some matrix completion methods to interpolate the missing values by minimizing the rank of matrix. This solution can quantify both global temporal and spatial correlations among sensor readings but is still limited to capture the one-dimensional spatial similarity due to a fact that, in the matrix formulation, the sensors with two-dimensional spatial coordinates are mapped into a one-dimensional vector, unavoidably resulting in the spatial information loss [7].

Recently, a multi-view learning based method is introduced to capture both local and global information in terms of spatial and temporal perspective, achieving state-of-the-art performance [43]. It also demonstrates that both local and global spatial/temporal correlations play an important role in the data reconstruction. However, it introduces four different models to capture the local and global spatial and temporal information respectively and then a linear regression model is adopted to estimate the final missing values, resulting in a labor-intensive parameter tuning process. Moreover, it requires a supervised model training using a large non-missing dataset, which is impractical due to that the collected sensory data may already suffer certain reading loss.

As a result, in this paper, we aim to explore - *whether we can accurately recover the missing sensor values by* ***capturing the global multi-dimensional spatial-temporal correlations*** *using a model that only needs to* ***tune very few parameters*** *and does* ***not require any supervised training***. To solve this problem, different from previous works, we formate the spatial-temporal sensor data as *tensor* - a multi-dimensional extension of a matrix and introduce a tensor based recovery method. Nevertheless, applying this high-level idea into practice requires addressing several challenges. First, how to accurately map the sensors' 2-D coordinates into a matrix is a non-trivial problem, especially considering that the sensors deployed in physical world is not naturally as a square or rectangle array (*e.g.,* some places have no sensor deployed, but other locations have many sensors). Moreover, *tensor completion* can be formulated to solve the problem of minimization on the tensor rank. This general optimization problem is NP-hard and thus untraceable [23]. So how to approximate the tensor rank and efficiently solve the optimization problem while guaranteeing its convergence is also a challenging issue [15], especially for a large-scale real-world sensor dataset.

To address above challenges, we first map the sensors' geo-locations into a matrix by finding each sensor's $k$ nearest neighbors in terms of longitudes and latitudes by proposing a Nearest Neighbor (NN) based heuristic searching method. Moreover, instead of using one time-dimension to capture the temporal information, we propose to model the temporal feature in a multi-dimensional view by measuring the the periodic patterns in sensor data[1]. Figure 1 details the our general idea of using high-order tensor to formulate

[1]For example, the sensor values in same hours of a day or same day in a week are similar, so we can also model the temporal feature as a matrix, being similar to the spatial one.

the sensor readings. Furthermore, by taking the recent advance of tensor decomposition theory, we introduce a t-SVD (Tensor Singular Value Decomposition) [17] based tensor recovery method, which substantially transforms the optimization of tensor's tubal-rank into a tensor multi-rank minimization in the calculation-efficient Fourier domain. Then we further relax the $\ell_1$ norm minimization of tensor's multi-rank into its nuclear norm and finally recover the missing sensor data (see Section 4). In a nutshell, our main contributions are as follows:

- We propose a NN-based heuristic searching method to map the sensors with irregular geo-locations into a matrix through iteratively searching the spatially nearest neighbor for each sensor.
- We introduce a tensor completion based method to recover the missing values by capturing the spatial and temporal information in a multi-dimensional way. It only requires to tune one key parameter and without requiring non-missing training data.
- We introduce an efficient t-SVD based optimization scheme to solve the tensor completion problem with a theoretical guarantee of convergence to optimal solution. The experiments in several real-world sensory data demonstrate that our recovery accuracy outperforms other state-of-the-art approaches.

## 2 PRELIMINARY ANALYSIS

In this section, we will conduct a series of preliminary analysis using one-year (364 days) PM2.5 dataset recorded by 36 PM2.5 sensor stations in Beijing, sampling interval is 1-hour [43]. Through those experiments, we demonstrate that air quality dataset exhibits *Temporal Similarity*, *Spatial Similarity* and *Periodic Pattern*, which substantially reveals the intuitions behind our tensor-based recovery method.

### 2.1 Temporal Similarity

To measure the temporal similarity, we calculate the relative differences of sensor readings in two adjacent sampling time for all the 36 sensors in a whole year. As Figure 2 (a) shows, about 90% of the relative temporal differences are below 0.05 and 99% of the reading changes are less than 0.16, which indicates that a strong correlation exits in the time dimension.

### 2.2 Spatial Similarity

Shown by Figure 2 (b), we compare the readings of Sensor13 with its 7 nearest neighbor sensors (based on the Euclidean distance) in the same sampling time. We can see that, for its top-3 spatial nearest sensors, more than 90% of the relative differences are less than 0.1 and 99% of the differences are below around 0.25, showing real PM2.5 data have strong spatial correlations.

### 2.3 Periodic Pattern

As we know, sensory data usually exhibits similar environmental behaviors at the same time of different days, or same day in different weeks such as PM2.5 reading will be higher in the traffic busy time (*e.g.,* daily commuting travel time) and lower in the night time. In our experiments, similar to the spatial and temporal case, we observe significant periodic pattern (around 70% of the differences are below 0.142, and 90% of the differences are less than0.278), which is shown by Figure 2 (c).

As a result, these three major characteristics - *temporal similarity* and *spatial similarity* as well as *periodic pattern*, highly motivate why we use a multi-dimensional tensor to unfold the spatial and temporal information and why we minimize the tensor multi-rank in the Fourier domain.

## 3 MAPPING THE IRREGULAR GEO-TAGGED SENSORS INTO AN ARRAY

As we analyzed before, the first challenge of applying tensor-based recovery method into practical geo-tagged time-series data is how to map those irregularly deployed sensors into an array. To well preserve the *Spatial Similarity*, the adjacent sensors in practice should also be near to each other in the constructed sensor array. This substantially meets the *First Law of Geography*, namely, *Everything is related to everything else, but near things are more related than distant things*, which also coincides with our spatial similarity observation in Section 2.

Basically, we can transform this sensor mapping problem into an Euclidean Distance minimization problem. We assume that $N$ geo-tagged sensors (labeled as $s_1, s_2, ..., s_N$) are mapped into a $n_1 \times n_2$ array (obviously $n_1 \times n_2 = N$) that is represented by matrix $S$. Each element $s_{i,j} (i = 1, 2, ..., n_1; j = 1, 2, ..., n_2)$ in the matrix indicates a sensor. Assuming that all the adjcent sensors of sensor $s_{i,j}$ is contained in the sensor-set $\mathcal{N}(s_{i,j})$ and the number of sensors in this set is represented as $|\mathcal{N}(s_{i,j})|$, then we can formulate the sensor mapping problem as the following minimization problem:

$$S^* = \arg \min_{S \in \mathcal{S}(N)} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{s_k \in \mathcal{N}(s_{i,j})}^{|\mathcal{N}(s_{i,j})|} Dis(s_{i,j}, s_k) \tag{1}$$

where $S^*$ indicates the optimal matrix we constructed, $s_k$ represent a sensor in sensor set $\mathcal{N}(s_{i,j})$, and $Dis(*, *)$ means the Euclidean distance of two sensors. $\mathcal{S}(N)$ is a matrix set that captures all the possible matrices that each element is randomly picked up from sensors $\{s_1, s_2, ..., s_N\}$.

Actually, the above optimization problem is untraceable since the $|\mathcal{S}(N)| = N!$, the permutation of $N$. So the complexity of solving this optimization problem using an exhausted searching is $O(N!)$. To simplify this problem, in this paper we propose a nearest neighbor based heuristic search method. The intuition behind is that although it is difficult for us to find the optimal solution for Eqn. 1 we can easily know the sensors in the very first left should be mapped into the first column of the matrix. Then given the sensors in the first column, we can heuristically search a sensor from the remaining sensors that has a minimum overall distance with its known adjacent sensors and put it in the next empty location in the matrix. By iteratively doing so, we can gradually narrow down the candidate sensors for the remaining locations in the matrix and make sure each sensor in the matrix has picked up the nearest neighbor into its adjacent locations. Since this is a heuristic approach based on the nearest neighbor searching, we cannot prove it yields the optimal solution theoretically. It actually gives us a sub-optimal result but with a relatively low computation overhead. The pseudo-code is shown in Algorithm 1.

Taking the 36 sensor stations in the air-quality dataset as an example, we first find the 4 nearest neighboring sensors (for Sensor-34) along the longitude (put those sensors into the column of a

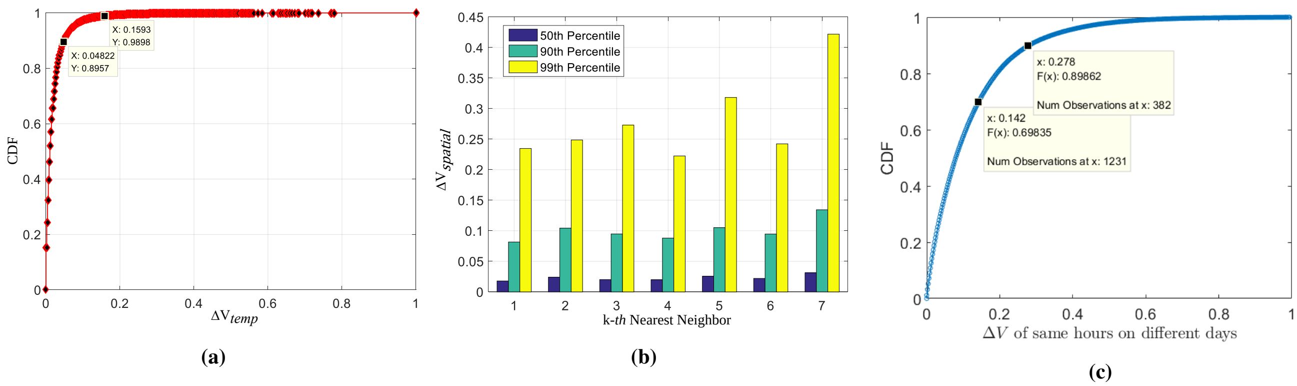
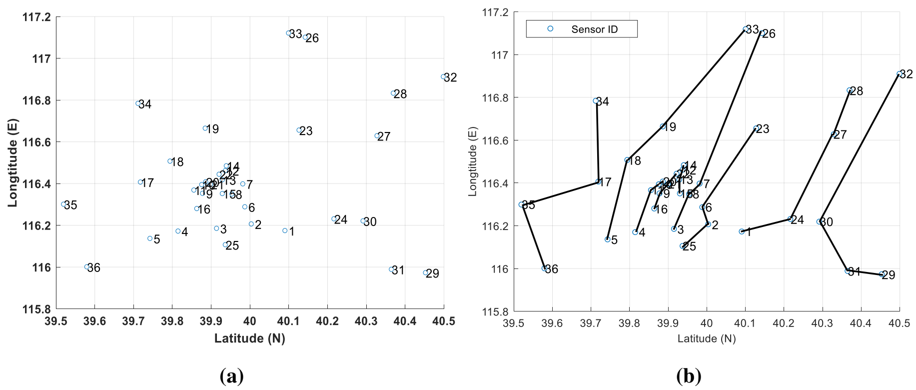



**Figure 2: (a) CDF of relative temporal differences of adjacent hours; (b) CDF of relative spatial differences of Sensor13 and its nearest neighbors; (c) CDF of relative differences in periodic pattern for same hours on different days**




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| S34 | S33 | S20 | S22 | S14 | S26 | S23 | S28 | S32 |
| S17 | S19 | S10 | S21 | S12 | S7 | S6 | S27 | S30 |
| S35 | S18 | S11 | S9 | S13 | S8 | S2 | S24 | S31 |
| S36 | S5 | S4 | S16 | S15 | S3 | S25 | S1 | S29 |

(c)

**Figure 3: (a) The spatial locations of PM2.5 sensor stations throughout the city; (b) The searching result by using NN-based heuristic searching method; (c) The mapped matrix of sensor array**

matrix) and then we consider the next nearest sensor along the latitude and formulate it as the second column of a matrix. By iteratively doing so, we finally can fill in a $4 \times 9$ matrix using all those sensors, which eventually model the spatial similarity via a 2-dimensional matrix. Figure 3 (a)~(c) show an example of how our NN-based Heuristic Searching method maps irregularly deployed 36 PM2.5 sensors into a sensor array.

Apart form the spatial similarity, we also need to model the temporal similarity and periodic patterns (as analyzed in Section 2). The most straightforward way is to formulate as one-dimensional array (plus the spatial dimensions, forming a $4\times9\times8759$ data tensor), or we can formulate as two dimensions to capture the similarity of same hours in a day (overall we can form a $4 \times 9 \times 24 \times 364$ 4-order tensor), or as three dimensions to model the similarities of same hours during different days and same days during different weeks (overall forming a $4\times9\times24\times7\times52$). The general idea is also shown via Figure 1.

In the next, given the formulated data tensor, we will elaborate how to use a tensor-SVD based recovery method to estimate those missing values.

# 4 TENSOR SVD BASED SENSOR DATA RECOVERY

In this section, we will briefly introduce the notations and definitions that are used in our method. For simplicity, all the formulation, mathematical theorems and definitions are based on a 3-order tensor, which can be naturally extended into high-order tensor cases.

We represent matrices by upper letters ($A$) and a $d$-order tensor is written by calligraphic letters ($\mathcal{A}$). $\mathcal{A}(i, j, k)$ denotes the $(i, j, k)$-th element of third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{A}(i, j, :)$ denotes the $(i, j)$-th tubal scalar. $\mathcal{A}(i, :, :)$, $\mathcal{A}(:, j, :)$, $\mathcal{A}(:, :, k)$ (or equivalently $\mathcal{A}^{(k)}$) denote the $i$-th horizontal slice, $j$-th lateral slice and $k$-th frontal slice. The $\hat{\mathcal{X}} = \text{fft}(\mathcal{X}, [\,], i)$ denotes the FFT on the $i$-th dimension of a multi-way array [47].

We then introduce following related definitions.

DEFINITION 1. *t-product: given two third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2\times n_4\times n_3}$, the t-product $C = \mathcal{A} * \mathcal{B}$ is a tensor of size $n_1 \times n_4 \times n_3$ given by $C = \mathcal{A} * \mathcal{B} = Fold(bcirc(\mathcal{A}) \cdot Unfold(\mathcal{B}))$. The $bcirc(\mathcal{A})$ is block circulant matrix and its first column is $[A^{(1)^T}, A^{(2)^T}, \cdots, A^{(n_3)^T}]$. The $Unfold(\cdot)$ and $Fold(\cdot)$*

**Algorithm 1:** NN-based Heuristic Searching for Sensor-Array Mapping

**Input:** Sensor Set: $\mathcal{S}_N = \{s_1, s_2, ..., s_N\}$
Mapping Marix: $S \in \mathbb{R}^{n_1 \times n_2}$
Manually pick up $n_1$ sensors filling in $S_{*,1}$
$\mathcal{S}_N \leftarrow \mathcal{S}_N - \{S_{*,1}\}$
**for** $i = 1 : n_1$ **do**
  **for** $j = 2 : n_2$ **do**
    **if** $i == 1$ **then**
      $S_{i,j} = \{s_k | \min_{s_k} \sum \mathbf{Dis}(s_k, \{S_{i,j-1}, S_{i+1,j-1}\})\}$
    **end**
    **else if** $i > 1$ && $i < n_1$ **then**
      $S_{i,j} = \{s_k | \min_{s_k} \sum \mathbf{Dis}(s_k, \{S_{i-1,j}, S_{i-1,j-1}, S_{i,j-1}, S_{i+1,j-1}\})\}$
    **end**
    **else**
      $S_{i,j} = \{s_k | \min_{s_k} \sum \mathbf{Dis}(s_k, \{S_{i-1,j}, S_{i-1,j-1}, S_{i,j-1}\})\}$
    **end**
    $\mathcal{S}_N \leftarrow \mathcal{S}_N - S_{i,j}$
  **end**
**end**
**Output:** Mapping Matrix $S$, indicating the sensor locations

*operators mean that Unfold*$(\mathcal{B}) = [B^{(1)^T}, B^{(2)^T}, \cdots, B^{(n_3)^T}]$ *and Fold*$(Unfold(\mathcal{B})) = \mathcal{B}$.

It is consistent with the multiplication of matrices if $n_3 = 1$.

Definition 2. *Identity tensor: the identity tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is a tensor whose first frontal slice is the $n_1 \times n_1$ identity matrix and all other frontal slices are zero.*

Definition 3. *Orthogonal tensor: a tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is orthogonal if $\mathcal{Q} * \mathcal{Q}^* = \mathcal{Q}^* * \mathcal{Q} = \mathcal{I}$.*

Definition 4. *f-diagonal tensor: a tensor is called f-diagonal if each frontal slice of the tensor is a diagonal matrix.*

## 4.1 Tensor Singular Value Decomposition

Given the definition of t-product, we introduce the tensor Singular Value Decomposition (t-SVD) [16].

Theorem 1. *For a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it can be factored as $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$, where $\mathcal{U}$ and $\mathcal{V}$ are orthogonal tensors of size $n_1 \times n_1 \times n_3$ and $n_2 \times n_2 \times n_3$ respectively. $\mathcal{S}$ is a rectangular f-diagonal tensor of size $n_1 \times n_2 \times n_3$.*

Definition 5. *The diagonal of $F_n circ(v) F_n^* = fft(v)$, where $fft(v)$ is the result of applying the Fast Fourier Transform to $v$, i.e. $diag(F_n circ(v) F_n^*) = fft(v)$.*

In Definition 1, the t-product is defined by circulant convolution, the computation of t-SVD can be efficiently calculated using the fast Fourier transform (FFT) [17]. For a 3-order tensor, we first apply the FFT along the third dimension to attain the Fourier transformed tensor $\hat{\mathcal{X}}$, and then compute the standard matrix SVD of each frontal slice of $\hat{\mathcal{X}}$. Finally, we apply an inverse FFT to the third dimension of the component tensors to compute the final t-SVD decomposition. For the higher order tensors, this concept of the t-SVD can be recursively extended by the t-product [28]. For details about this process, see the t-SVD in Algorithm 2.

**Algorithm 2:** t-SVD

**Input:** $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$, $\gamma = n_3 n_4 \cdots n_N$
**for** $i = 3 : N$ **do**
  $\hat{\mathcal{X}} \leftarrow$ fft$(\mathcal{X}, [\,], i)$;
**end**
**for** $i = 1 : \gamma$ **do**
  $[\hat{U}, \hat{S}, \hat{V}] = \mathrm{SVD}(\hat{\mathcal{X}}(:,:,i))$;
  $\hat{\mathcal{U}}(:,:,i) = \hat{U}$; $\hat{\mathcal{S}}(:,:,i) = \hat{S}$; $\hat{\mathcal{V}}(:,:,i) = \hat{V}$;
**end**
**for** $i = 3 : N$ **do**
  $\mathcal{U} \leftarrow$ ifft$(\hat{\mathcal{U}}, [\,], i)$; $\mathcal{S} \leftarrow$ ifft$(\hat{\mathcal{S}}, [\,], i)$; $\mathcal{V} \leftarrow$ ifft$(\hat{\mathcal{V}}, [\,], i)$;
**end**
**Output:** ( $\mathcal{U}, \mathcal{S}$, and $\mathcal{V}$ )

The t-SVD of the 3-order tensor are shown in Figure 4. The construction of the t-SVD is similar to the matrix SVD $X = USV^T$ except that the $t$-product and tensor transpose substitute by the equivalent matrix operations [10]. Similar to the matrix SVD, the t-SVD can also be written as the sum of outer $t$-products.

Based on the t-SVD, we can define the notion of the tensor rank as follows:

Definition 6. *Tensor multi-rank: the multi-rank of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a vector $\mathbf{r} \in \mathbb{R}^{n_3}$ with the i-th element equal to the rank of the i-th frontal slice of $\hat{\mathcal{A}}$ obtained by taking the Fourier transform along the third dimension of the tensor, i.e. $\mathbf{r}_i = rank\hat{\mathcal{A}}(:,:,i)$.*

Definition 7. *Tensor tubal-rank: the tensor tubal rank of a 3-D tensor is defined to be the number of non-zero tubes of $\mathcal{S}$ in the t-SVD factorization.*

Theorem 2. *For a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor nuclear norm (TNN) is defined as the sum of the singular values of all the frontal slices of $\hat{\mathcal{A}}$, denoted by $\|\mathcal{A}\|_{TNN}$, which is the tightest convex relaxation to $\ell_1$ norm of the tensor multi-rank, i.e. $\|\mathcal{A}\|_{TNN} = rank(blkdiag(\hat{\mathcal{A}}))$.*

Here, the blkdiag$(\hat{\mathcal{A}})$ is a block diagonal matrix defined as follows:

$$\mathrm{blkdiag}(\hat{\mathcal{A}}) = \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(n_3)} \end{bmatrix} \tag{2}$$

Where $\hat{\mathcal{A}}^{(i)}$ is the $i$-th frontal slice of $\hat{\mathcal{A}}$, $i = 1, 2, ..., n_3$.

## 4.2 Problem Formulation

First, we mathematically define our target problem. Assuming that we have $n_1 \times n_2$ sensors deployed in different spatial areas and collect sensor readings for $T$ timestamps (see the example in Figure 1), we then can formulate it as a 3-order tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times T}$. We define a projection operator $\mathcal{P}_\Omega(\mathcal{M}) : \mathbb{R}^{n_1 \times n_2 \times T} \to \mathbb{R}^K, \Omega \in \{0,1\}^{n_1 \times n_2 \times T}$ that indicates the $K$ observed sensor readings. Hence our goal is to accurately recover the true sensor readings $\mathcal{X}$ from a partially observed data tensor $\mathcal{M}_\Omega$.

Being similar to matrix completion, this problem can be formulated as solving a low-rank minimization problem:

$$\min \operatorname{rank}_t(\mathcal{X}) \text{ s.t. } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}) \tag{3}$$

where $\operatorname{rank}_t(\mathcal{X})$ is the tubal rank. However, tensor tubal-rank is NP-hard [46]. Thus, to make it tractable, we replace tubal-rank by a relaxation convex surrogate tensor nuclear norm (TNN) as follows

$$\min \|\mathcal{X}\|_{TNN} \text{ s.t. } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}) \tag{4}$$

From Theorem 2, by leveraging the definition of $\|\mathcal{X}\|_{TNN} = \|\operatorname{blkdiag}(\hat{\mathcal{X}})\|_*$ (where $\|\cdot\|_*$ denotes the nuclear norm of matrix, *i.e.,* the sum of its singular values), Eqn. (4) is equivalent with the following equivalent form:

$$\min \|\operatorname{blkdiag}(\hat{\mathcal{X}})\|_* \text{ s.t. } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}) \tag{5}$$

To resolve the dependence between the frontal slices of the $\hat{\mathcal{X}}$, we introduce ADMM [2] to split these interdependent terms. Specifically, by introducing an additional tensor $\mathcal{Z}$, we reformulate (5) equivalently as follows:

$$\min \|\operatorname{blkdiag}(\hat{\mathcal{Z}})\|_* \text{ s.t. } \hat{\mathcal{X}} - \hat{\mathcal{Z}} = 0,\ P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}) \tag{6}$$

### 4.3 ADMM for Solving Tensor Completion

To solve the optimization problem Eqn. (6), we first introduce the partial augmented Lagrangian function of Eqn. (6) as below.

$$\mathcal{L}_\mu(\hat{\mathcal{Z}}, \hat{\mathcal{X}}, \hat{\mathcal{W}}) = \|\operatorname{blkdiag}(\hat{\mathcal{Z}})\|_* + \langle \hat{\mathcal{W}}, \hat{\mathcal{X}} - \hat{\mathcal{Z}} \rangle + \mu/2 \|\hat{\mathcal{X}} - \hat{\mathcal{Z}}\|_F^2 \tag{7}$$

where $\mathcal{W}$ is the Lagrange multipliers and $\mu$ is a penalty parameter. We present an alternating direction method of multipliers (ADMM) iterative optimization scheme to successively minimize $\mathcal{L}_\mu$ over $(\mathcal{Z}, \mathcal{X})$ and then update $\mathcal{W}$ as follows.

**Update** $\mathcal{Z}^{k+1}$: Firstly, we fix $\mathcal{X}$ to optimize $\mathcal{Z}$ by solving:

$$\hat{\mathcal{Z}}^{k+1} = \arg\min_{\hat{\mathcal{Z}}} \|\operatorname{blkdiag}(\hat{\mathcal{Z}})\|_* + \mu/2 \|\hat{\mathcal{Z}} - (\hat{\mathcal{X}}^k + 1/\mu \hat{\mathcal{W}}^k)\|_F^2 \tag{8}$$

According to the definition of TNN, we can solve each frontal slice $\hat{\mathcal{Z}}^{k+1,(i)}, i = 1, ..., n_3$ by splitting problem (8) into $n_3$ independent minimization problems. Then the resulting each subproblem with respect to $Y = \hat{\mathcal{Z}}^{k+1,(i)} \in \mathbb{R}^{n_1 \times n_2}$ is formulated as follows:

$$\hat{\mathcal{Z}}^{k+1,(i)} = \arg\min_Y \|Y\|_* + \frac{\mu}{2} \|Y - (\hat{\mathcal{X}}^{k,(i)} + 1/\mu \hat{\mathcal{W}}^{k,(i)})\|_F^2 \tag{9}$$

Leveraging the *singular value thresholding (SVT)* operator for a matrix [3], we can calculate each $\hat{\mathcal{Z}}^{k+1,(i)}$ by

$$\hat{\mathcal{Z}}^{k+1,(i)} = \hat{U} D_{1/\mu}(\hat{S}) \hat{V}^T = \hat{U} \operatorname{diag}(\hat{\mathcal{S}}(i,i,:) - 1/\mu) \hat{V}^T \tag{10}$$

where $\operatorname{SVD}(\hat{\mathcal{X}}^{k,(i)} + 1/\mu \hat{\mathcal{W}}^{k,(i)}) = \hat{U}\hat{S}\hat{V}^T$, $(\mathcal{X}^k + 1/\mu \mathcal{W}^k) = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ and $D_{1/\mu}(\hat{S}) = \operatorname{diag}(\hat{S}(i,i,:) - 1/\mu)_+$, where $t_+ = max(0, t)$, i.e. the positive part of $t$.

If we define t-SVD of $\mathcal{Z}^{k+1}$ as $\mathcal{Z}^{k+1} = \mathcal{U} * \mathcal{D} * \mathcal{V}^T$, then the solution for Eqn.(8) is given by $\hat{\mathcal{D}}(:,:,i) = \operatorname{diag}(\hat{\mathcal{S}}(i,i,:) - 1/\mu_+)$ in Fourier domain, and we can use the inverse FFT to recover $\mathcal{Z}$ in original domain [22].

**Update** $\mathcal{X}^{k+1}$: To update $\mathcal{X}^{k+1}$, we have the following subproblem:

$$\begin{aligned} \mathcal{X}^{k+1} &= \arg\min_{\mathcal{X}} \frac{\mu}{2} \|\mathcal{X} - \mathcal{Z}^{k+1} + 1/\mu \mathcal{W}^k\|_F^2 \\ &\text{s.t. } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}) \end{aligned} \tag{11}$$

According Karush-Kuhn-Tucker (KKT) conditions, the solution of this function (11) is $\mathcal{X}^{k+1} := P_\Omega(\mathcal{M}) + P_{\bar{\Omega}}(\mathcal{Z}^{k+1} - 1/\mu \mathcal{W}^k)$, where $\bar{\Omega}$ represents the supplementary set of $\Omega$.

**Update** $\mathcal{W}^{k+1}$: Last, the Lagrange multipliers is updated by $\mathcal{W}^{k+1} = \mathcal{W}^k + \mu(\mathcal{X}^{k+1} - \mathcal{Z}^{k+1})$. Based on the above analysis, we develop an ADMM algorithm for the tensor-SVD and completion problem (4), as outlined in Algorithm 3.

**Algorithm 3:** ADMM for sensor data completion based on t-SVD

**Input:** $\mathcal{M}$, $\Omega$
**Initialization:**
$\mathcal{X}^0 = \mathcal{Z}^0 = \mathcal{W}^0 = 0, \mu = 0.001, \epsilon_1 = 10^{-6}, \epsilon_2 = 10^{-4}$
**while** *not converged* **do**
  Update the each slice of $\hat{\mathcal{Z}}^{k+1}$ by Eqn. (10):
  Update $\mathcal{X}^{k+1}$ by Eqn. (11);
  Update $\mathcal{W}^{k+1}$ by $\mathcal{W}^{k+1} = \mathcal{W}^k + \mu(\mathcal{X}^{k+1} - \mathcal{Z}^{k+1})$;
  Check the convergence condition,
  $\|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_F\} < \epsilon_1, \|\mathcal{Z}^{k+1} - \mathcal{X}^k\|_F\} < \epsilon_2$
**end**
**Output:** ( $\mathcal{X}$ )

This algorithm can also be accelerated by adaptively changing for the Lagrangian parameter $\mu$, increasing $\mu^k$ iteratively by $\mu^{k+1} = \rho\mu^k$, where $\rho \in (1.0, 1.1]$ in general and $\mu_0$ is a very small constant.

## 5 EXPERIMENTAL RESULTS

In this section, we will show the experimental results by comparing with other state-of-the-art missing data recovery methods, including filter and regression based approaches and tensor/matrix based recovery methods.

### 5.1 Datasets and Evaluation Metrics

**Datasets and Evaluation Metrics:** In the paper, we test our method using several real-world datasets: air quality and meteorological data in Beijing. The air quality dataset has 8,759 PM2.5 readings collected from 2014-05-01 to 2015-04-30 by 36 monitoring stations, the sampling interval is 1 hour. The overall missing ratio of PM2.5 readings is 13.25% in the air quality dataset, which contains 8.15% general missing and 2.15% spatial block missing[2]. The meteorological dataset contain six different types of data recorded by 16 sensors throughout the Beijing City including in CO, NO2, Humidity and Wind-Speed as well Wind-Direction. To form the ground truth dataset, we choose the missing sensor readings using a same scheme as described in [43].

[2]Spatial block missing means at certain sampling times, the readings from all sensors are missing.

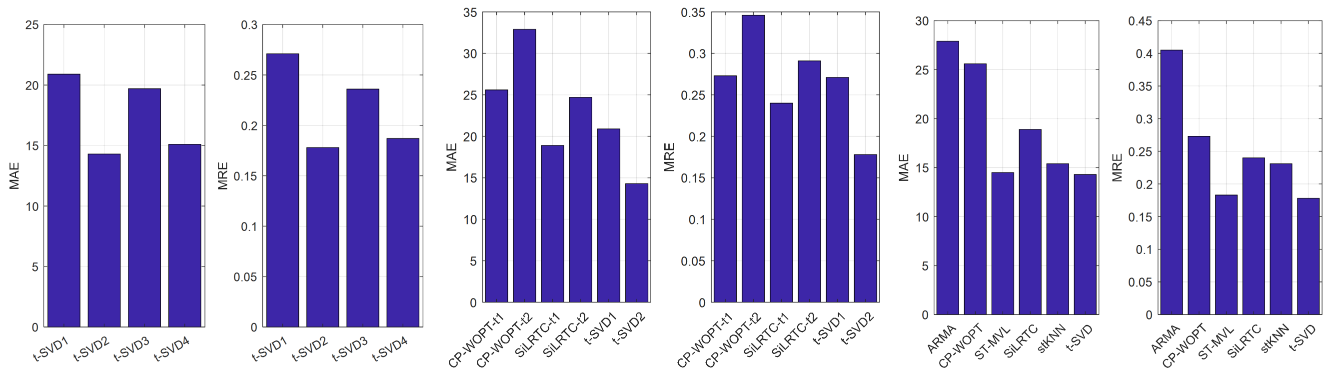



**Figure 4: (a) Recovery accuracies for different tensor formulations; (b) Comparison of recovery accuracies for different tensor-based methods; (c) Comparison of state-of-the-art sensor reading recovery methods**

We adopt the standard Mean Absolute Error (MAE) and Mean Relative Error (MRE) as the evaluation metrics [43].

$$\text{MAE} = \frac{\sum_i^m |v_i - \hat{v}_i|}{m}; \ \text{MRE} = \frac{\sum_i^m |v_i - \hat{v}_i|}{\sum_i^m v_i} \tag{12}$$

where $\hat{v}_i$ is the predicted value and $v_i$ is the ground truth and $m$ is the overall number of the missing readings.

## 5.2 Baseline Methods

We compare our method with other typical sensor data recovery methods:

*i)* ARMA (AutoRegressive Moving-Average): it is one of the most popular models for predicting time series data. ARMA has two polynomial terms, one is for modeling auto-regression process and another one is performing moving average.

*ii)* stKNN (spatial and temporal K-Nearest Neighbors): it adopts the $k$ nearest spatial and temporal neighbors as a prediction.

*iii)* ST-MVL [43]: it is the newest work that achieves the state-of-the-art performance on missing sensor reading recovery. It is built upon a multi-view learning framework.

*iv)* CP-WOPT [1]: it solves a weighted least squares problem based on the CANDECOMP/PARAFAC (CP) decomposition.

*v)* SiLRTC [24]: it uses a low-n-rank tensor nuclear norm to approximate the tensor rank.

The first three approaches are typical works concentrating on missing sensor data recovery, and the latter two methods are tensor completion methods. However, since those tensor-based techniques are primarily designed for recovering noisy images or videos (naturally can be seen as a tensor) and cannot directly applied to our PM2.5 and meteorological datasets, we first use our NN-based heuristic searching method to transform the irregularly deployed sensors into an array and then feed into those methods. The parameter in above baseline methods are tuned based on the tuning description or parameter settings in corresponding papers.

## 5.3 Experimental Settings

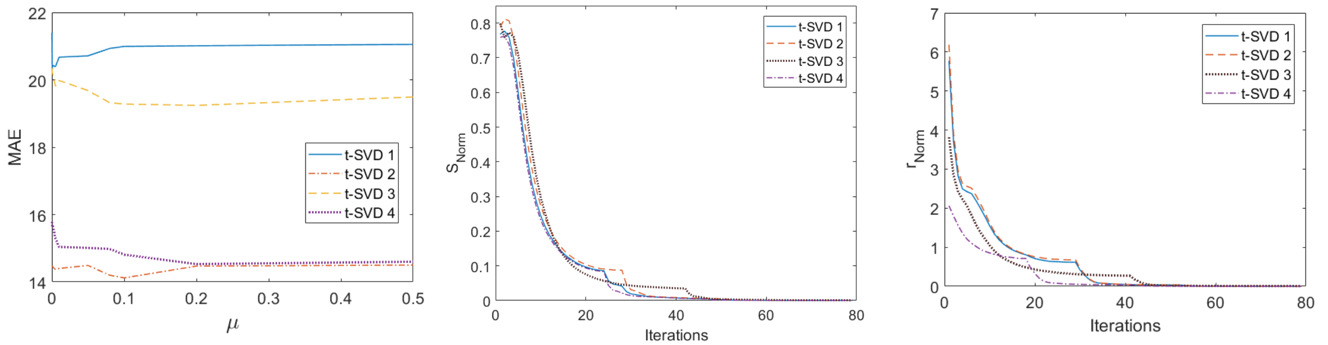
As analyzed before, to capture the spatial and temporal correlations, we formulate the sensor data as four tensors with different sizes: t-SVD 1: $36\times24\times364$, t-SVD 2: $4\times9\times8736$, t-SVD 3: $4\times9\times24\times364$ and t-SVD 4: $4\times9\times168\times52$. The t-SVD 1 only formulates the spatial locations of sensors as a vector, and the other three formulations otherwise model the spatial feature as a matrix as per Section 3.

For the time dimension, we formulate it as *Hour* $\times$ *Day* $(24\times364)$ or *Hour* $\times$ *Week* $(168 \times 52)$ or a 1-D vector. We set the related parameters as: $\mu = 0.001, \epsilon_1 = 10^{-6}, \epsilon_2 = 10^{-4}$ and maximum iteration number as 500. In our model, the only parameter we need to tune is the penalty parameter $\mu$ ($\epsilon_1, \epsilon_2$ are stop conditions and being manually set without tuning in this paper).

## 5.4 Results in Air Quality Dataset

In the section, we report the experimental results via Figure 4 (a)~(c). In Figure 4 (a), we compare the recovery performance in terms of different tensor formulation sizes. We can see that formulating the data as a $4 \times 9 \times 8736$ tensor or $4 \times 9 \times 168 \times 52$ tensor achieves a better result. Then we compare our results with other tensor based techniques in terms of two tensor formulations (*i.e.,* $36 \times 24 \times 364$ and $4 \times 9 \times 8736$). We find that, comparing to other popular tensor completion methods (normally adopted in recovering the visual data), t-SVD also exhibits a better result. This is due to that CP-WOPT highly relies on the CP rank which is difficult to accurately estimate especially for a real-world sensor dataset. For SiLRTC, it is based upon an assumption that the strong low tensor-rank characteristic exits in every mode of the recovered tensor which is also hard to satisfy for a practical sensor dataset.

In Figure 4 (c) we compare our solution with all the baseline methods (by picking corresponding best tensor formulation for the tensor based recovery solutions) overall our performance is the best in terms of MAE and MRE, especially much better than the ARMA and stKNN. The experimental results reveal that by formating the spatial information as a matrix our method can substantially preserves a 2-D relative geometry relations among different sensors and is more advance on capturing the latent global spatial correlation. It's worth mentioning that ST-MVL also achieve a comparable




(a) (b) (c)

**Figure 5: (a) Impact of parameter $\mu$ to the recovery accuracy; (b) Convergence of our model in terms of $s_{Norm}$; (c) Convergence of our model in terms of $r_{Norm}$;**

performance as our method since it also intensively considers the local/global temporal and spatial dependency of the sensors. But the superiority of t-SVD lies on that it only needs to tune very few parameters and does not require the non-missing training data.

## 5.5 Impact of Parameters and Convergence

This section explores the impact of parameters to the performance of the proposed recovery method. Figure 5(a) depicts the influence of parameter $\mu$ to the recovery accuracy in terms of MAE. Actually, parameter $\mu$ is the only key parameter that needs to be fine-tuned in the t-SVD methods. Through the experiments, we observe that the performance on missing data recovery is relatively robust to the changes of $\mu$, which are different from the baseline methods that are normally sensitive to the parameters (especially the regression based methods). Figure 5 (b)~(c) show the convergence of our model in terms of $s_{Norm} = \|\mathcal{Z}^{k+1} - \mathcal{X}^k\|_F$ and $r_{Norm} = \|\mathcal{Z}^{k+1} - \mathcal{Z}^k\|_F$, illustrating that t-SVD can fast converge to an optimal solution by about 40 $\sim$ 70 iterations.

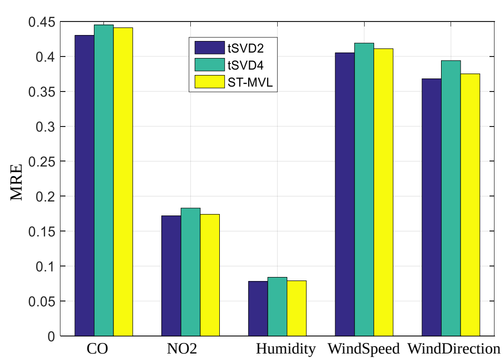



**Figure 6: Comparing our method with ST-MVL on five meteorological datasets in terms of MRE**

## 5.6 Results on Meteorological Datasets

This section reports the experimental evaluations on five meteorological datasets including CO, NO2, Humidity, Wind-Speed and Wind-Direction. As Figure 6 shows, we compare the results of t-SVD 2 and t-SVD 4 with ST-MVL in terms of MRE errors. By formulating the data as a 3-D tensor, our t-SVD method outperforms ST-MVL, the newest and most competitive work in sensor data recovery, in all five datasets. Although the improvement is not significant (from 1% to 6.3% improvement), our tensor based recovery method involves less parameter tuning process, in which the only required fine-tuning parameter is $\mu$ and it is robust to different types of sensor data, as shown in Figure 5 (a). Most importantly, t-SVD also does not require a large-scale non-missing training data to perform a supervised learning. Because in our optimization objective function, our primary goal is to minimize the overall tensor rank in instead of minimizing the loss function with true sensor readings. In practical, the true sensor readings are hard to collect (*i.e.,* they are missing in the first place before you train a model).

Overall, we believe the proposed tensor-based method provides an alternative data-driven and less-laborious approach for accurately recovering the missing spatial-temporal sensor data.

# 6 RELATED WORK

Recovering miss sensor readings given the data observed is a non-trivial research problem. Many promising methods are proposed by researchers from different communities such as meteorology [8], data mining [37], sensor network as well as computer vision [14]. In this paper, we review the related works from two aspects. We will first discuss recent research advances on sensor data recovery from different communities. Then we will intensively discuss latest matrix and tensor completion methods which are more close to the approaches used in this paper.

## 6.1 Missing Sensor Data Recovery

The intuitive solutions to solve the sensor reading missing is to model the quantitative relations between the observed elements and the missing values based on some regression or filtering methods such as linear regression, Kriging, ARMA and Inverse Distance

Weighting (IDW) *etc.*. Those methods usually are computationally efficient and interpretable so they are widely adopted in different application domains. For example, in [29], the authors integrate IDW with Geographical Information System (GIS) to estimate the incomplete rainfall records. Wu *et al.* [41] feed different spatial attributes such as latitudes, longitudes and latitudes into a residual Kriging method to interpolate the missing monthly temperature data. Literature [45] intensively compares different missing value recovery methods in a Turkish meteorological dataset.

Apart from those methods that focus on modeling the spatial or temporal dependencies, some more sophistical approaches intend to capture both temporal and spatial correlations of sensor readings. For instance, DEMS, proposed by Gruenwald *et al.* [13], aims to handle missing values in the domain of Mobile Sensor Network (MSN), in which the authors first convert mobile sensor readings into virtual static sensor readings and then mine the spatial-temporal relationships among sensors to estimate the missing readings. In [31], Pan *et al.* propose a $k$-Nearest Neighbors based missing data estimation method, which first adopts linear regression model to capture the spatial correlation among different sensor nodes and then utilizes the data information of multiple neighbor nodes to estimate the missing data jointly, considering both temporal and spatial correlations of sensors. Moreover, in the domain of recommendation, many collaborative filtering based methods are proposed to fill in the missing values in the user or item matrix [26, 27, 40]. Some surveys intensively discuss the related literatures [25, 42].

Recently, Yi and Zheng *et al.* introduce ST-MVL, a spatial-temporal multi-view learning based learning method that first model the global spatial and temporal correlations via regression based methods and further incorporates IDW and collaborative filtering to capture the local spatial-temporal dependencies. ST-MVL achieve a state-of-the-art performance in terms of filling missing geo-tagged sensor readings. However, it ensembles five different models and each model requires to fine-tune several parameters, which is labor intensive. Moreover, ST-MVL is still limited to capture one-dimensional spatio and temporal information and fail to model high-dimension spatial features (*e.g.,*sensors with longitude, latitude and altitude) and periodic pattern in the time dimension. In this paper, we propose a Tensor-SVD based method that can overcome the above shortcomings and still achieve a comparable or better recovery accuracy.

## 6.2 Matrix and Tensor Completion Approaches

To capture the global information of targeted dataset, the “rank” of the matrix is a powerful tool and many matrix completion based method are proposed [39]. In [4] authors show that under some mild condition, most low-rank matrices can be perfectly recovered from an incomplete dataset by solving a simple convex optimization program. Chen *et al.* [6] investigate the problem of low-rank matrix completion where a large number of columns are arbitrarily corrupted. They showed that only a small fraction of the entries are needed in order to recover the low-rank matrix with high probability. Klopp *et al.* [18] study the optimal reconstruction error in the case that the observations are noisy and column-wise or element-wise corrupted. Although low-rank matrix completion methods have shown some promising characteristics and played an important role in missing data recovery, however, such methods cannot work well on recovering spatial-temporal sensor data with multi-dimensional spatial correlations [5].

As a result, many tensor-based recovery methods are recently proposed. Actually, in many practical applications, the recovered dataset can be naturally treated as tensor, a multi-dimensional extension of matrix [23, 32]. Generally, there are two state-of-the-art techniques used for tensor completion. One is the nuclear norm minimization, many pioneering similar works are emerged [11, 20, 33, 35] since Liu *et al.* [23, 24] first extended the nuclear norm of matrix (*i.e.,* the sum of all the singular values) to tensor. Later on, Gandy *et al.* [11] and Signoretto *et al.* [34] consider a tractable and unconstrained optimization problem of low-n-rank tensor recovery and adopt the Douglas-Rachford splitting method and ADMM method. Another popular technique is to utilize the tensor decomposition [21, 36, 44], *i.e.,* decomposing the $N$th-order tensor into another smaller $N$th-order tensor (*i.e.,* core tensor) and $N$ factor matrices. Generally, Tucker and CANDECOMP/PARAFAC Decomposition are the two most popular tensor decomposition frameworks [19], which thus results in two different definitions of tensor rank, *i.e.,* multi-linear rank that is defined straightforwardly as a weighted sum of nuclear norm of mode-$n$ matricizations, and CP rank which is defined by the minimum number of the rank-one term in CP decomposition. In [1], Acar *et al.* develop an algorithm called CP-WOPT (CP Weighted OPTimization) used a first-order optimization approach for dealing with missing value and has been testified to provide a good imputation performance. However, those tensor completion methods are normally applied in visual data and cannot directly deal with a case that the sensors are deployed irregularly in a 2-D or 3-D spaces. In this paper, we first propose a NN-based heuristic searching method for transforming the sensors into an matrix by finding each sensor's $k$ nearest neighbors in terms of longitudes and latitudes. Then we introduce a Tensor Singular Value Decomposition based tensor recovery method, which substantially transforms the optimization of tensor's tubal-rank into a Fourier domain. By doing so, we can achieve a state-of-the-art recovering accuracy but with less parameter-tuning and computation overhead.

# 7 DISCUSSION & CONCLUSION

In this section, we briefly discuss the pros and cons of our approach and point out some unsolved issues that are left for our future work.

First of all, our approach is built upon the assumption that every mode of the tensor is low tubal-rank. This assumption might be too strong in practice. To deal with this issue, a straight-forward solution is to add a priori factor/weight to penalize different modes with some priori knowledge. A method that can adaptively find the low tubal-rank modes and only minimize the modes where low tubal-rank exits also worths an investigation in the future. Moreover, in our model, we assume the observed data are free of noise, but in practice those data (that we can observe) may also be polluted by some unknown noise. Therefore, in the future, it is necessary to consider a robust model to tackle those cases with noisy observations.

In summary, this paper proposes a missing data recovery method by formulating the spatial-temporal sensor data as a multi-dimensional tensor. The main novelty of this paper stands on two sides. On the one side, we propose a nearest neighbor based heuristic search method that can formulate the high-dimensional spatial information as a matrix/tensor. On the other side, the introduced t-SVD method only requires to tune one key parameters in a unsupervised manner and is computationally efficient. The intensive experiments on several real-world sensor datasets demonstrate that the proposed method can accurately model spatial and temporal dependencies among sensors to enable a high performance on missing sensor data recovery.

# REFERENCES


[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 1 (2011), 41–56.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* 3, 1 (2011), 1–122.

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

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

[5] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis. 2013. Low-rank matrix recovery from errors and erasures. *Information Theory, IEEE Transactions on* 59, 7 (2013), 4324–4337.

[6] Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. 2011. Robust matrix completion with corrupted columns. *arXiv preprint arXiv:1102.2254* (2011).

[7] Zhe Chen and Andrzej Cichocki. 2005. Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep* 68 (2005).

[8] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. 2009. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons.

[9] Li Da Xu, Wu He, and Shancang Li. 2014. Internet of things in industries: a survey. *Industrial Informatics, IEEE Transactions on* 10, 4 (2014), 2233–2243.

[10] Gregory Ely, Shuchin Aeron, Ning Hao, and Misha E Kilmer. 2015. 5D seismic data completion and denoising using a novel class of tensor decompositions. *Geophysics* 80, 4 (2015), V83–V95.

[11] Silvia Gandy, Benjamin Recht, and Isao Yamada. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27, 2 (2011), 025010.

[12] Mohinder S Grewal. 2011. *Kalman filtering*. Springer.

[13] Le Gruenwald, Md Shiblee Sadik, Rahul Shukla, and Hanqing Yang. 2010. DEMS: a data mining based technique to handle missing data in mobile sensor network applications. In *Proceedings of the Seventh International Workshop on Data Management for Sensor Networks*. ACM, 26–32.

[14] Tamir Hazan, Simon Polak, and Amnon Shashua. 2005. Sparse image coding using a 3D non-negative tensor factorization. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Vol. 1. IEEE, 50–57.

[15] Christopher J Hillar and Lek-Heng Lim. 2013. Most tensor problems are NP-hard. *Journal of the ACM (JACM)* 60, 6 (2013), 45.

[16] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover. 2013. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.* 34, 1 (2013), 148–172.

[17] Misha E Kilmer and Carla D Martin. 2011. Factorization strategies for third-order tensors. *Linear Algebra Appl.* 435, 3 (2011), 641–658.

[18] Olga Klopp, Karim Lounici, and Alexandre B Tsybakov. 2014. Robust matrix completion. *arXiv preprint arXiv:1412.8132* (2014).

[19] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM review* 51, 3 (2009), 455–500.

[20] Nadia Kreimer, Aaron Stanton, and Mauricio D Sacchi. 2013. Tensor completion based on nuclear norm minimization for 5D seismic data reconstruction. *Geophysics* 78, 6 (2013), V273–V284.

[21] Daniel Kressner, Michael Steinlechner, and Bart Vandereycken. 2014. Low-rank tensor completion by Riemannian optimization. *BIT Numerical Mathematics* 54, 2 (2014), 447–468.

[22] Adrian S Lewis. 1995. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis* 2, 1 (1995), 173–183.

[23] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2009. Tensor completion for estimating missing values in visual data. In *Computer Vision, 2009 IEEE 12th Intl. Conference on*. 2114–2121.

[24] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2013. Tensor completion for estimating missing values in visual data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35, 1 (2013), 208–220.

[25] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision Support Systems* 74 (2015), 12–32.

[26] Hao Ma, Irwin King, and Michael R Lyu. 2007. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 39–46.

[27] Benjamin M Marlin, Richard S Zemel, Sam T Roweis, and Malcolm Slaney. 2011. Recommender Systems, Missing Data and Statistical Model Estimation.. In *IJCAI proceedings-international joint conference on artificial intelligence*, Vol. 22. 2686.

[28] Carla D Martin, Richard Shafer, and Betsy LaRue. 2013. An order-p tensor factorization with applications in imaging. *SIAM Journal on Scientific Computing* 35, 1 (2013), A474–A490.

[29] Mohamad J Noori, Hussein H Hassan, and Yaseen T Mustafa. 2014. Spatial Estimation of Rainfall Distribution and Its Classification in Duhok Governorate Using GIS. *Journal of Water Resource and Protection* 6, 02 (2014), 75.

[30] Anthony M Norcia, Maureen Clarke, and Christopher W Tyler. 1985. Digital filtering and robust regression techniques for estimating sensory thresholds from the evoked potential. *IEEE Engineering in Medicine and Biology Magazine* 4, 4 (1985), 26–32.

[31] Liqiang Pan and Jianzhong Li. 2010. K-nearest neighbor based missing data estimation algorithm in wireless sensor networks. *Wireless Sensor Network* 2, 02 (2010), 115.

[32] Wenjie Ruan, Peipei Xu, Quan Z Sheng, Nickolas JG Falkner, Xue Li, and Wei Emma Zhang. 2017. Recovering Missing Values from Corrupted Spatio-Temporal Sensory Data via Robust Low-Rank Tensor Completion. In *International Conference on Database Systems for Advanced Applications*. Springer, 607–622.

[33] Wenjie Ruan, Peipei Xu, Quan Z Sheng, Nguyen Khoi Tran, Nickolas JG Falkner, Xue Li, and Wei Emma Zhang. 2016. When sensor meets tensor: filling missing sensor values through a tensor approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2025–2028.

[34] Marco Signoretto, Lieven De Lathauwer, and Johan AK Suykens. 2010. Nuclear norms for tensors and their use for convex multilinear estimation. *Linear Algebra and Its Applications* 43 (2010).

[35] Marco Signoretto, Raf Van de Plas, Bart De Moor, and Johan AK Suykens. 2011. Tensor versus matrix completion: a comparison with application to spectral data. *Signal Processing Letters, IEEE* 18, 7 (2011), 403–406.

[36] Laurent Sorber, Marc Van Barel, and Lieven De Lathauwer. 2013. Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-(L_r,L_r,1) terms, and a new generalization. *SIAM Journal on Optimization* 23, 2 (2013), 695–720.

[37] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 374–383.

[38] Yufei Tao, Jimeng Sun, and Dimitris Papadias. 2003. Analysis of predictive spatio-temporal queries. *ACM Transactions on Database Systems (TODS)* 28, 4 (2003), 295–336.

[39] Kim-Chuan Toh and Sangwoon Yun. 2010. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization* 6, 615-640 (2010), 15.

[40] Jun Wang, Arjen P De Vries, and Marcel JT Reinders. 2006. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 501–508.

[41] Tingting Wu and Yingru Li. 2013. Spatial interpolation of temperature in the United States using residual kriging. *Applied Geography* 44 (2013), 112–120.

[42] Zhe Yang, Bing Wu, Kan Zheng, Xianbin Wang, and Lei Lei. 2016. A Survey of Collaborative Filtering-Based Recommender Systems for Mobile Internet Applications. *IEEE Access* 4 (2016), 3273–3287.

[43] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. 2016. ST-MVL: Filling Missing Values in Geo-sensory Time Series Data. In *IJCAI 2016*.

[44] Tatsuya Yokota, Qibin Zhao, Chao Li, and Andrzej Cichocki. 2015. Smooth PARAFAC Decomposition for Tensor Completion. *arXiv preprint arXiv:1505.06611* (2015).

[45] Ceylan Yozgatligil, Sipan Aslan, Cem Iyigun, and Inci Batmaz. 2013. Comparison of missing value imputation methods in time series: the case of Turkish meteorological data. *Theoretical and applied climatology* 112, 1-2 (2013), 143–167.

[46] Zemin Zhang and Shuchin Aeron. 2015. Exact tensor completion using t-SVD. *arXiv preprint arXiv:1502.04689* (2015).

[47] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 2014. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3842–3849.