

Received September 28, 2017, accepted October 31, 2017, date of publication November 7, 2017,
date of current version February 14, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2770146

Missing Data Recovery Based on Tensor-CUR Decomposition

LELE WANG¹, KUN XIE¹, THABO SEMONG¹, AND HUIBIN ZHOU²

¹College of Computer Science and Electronics Engineering, Hunan University, Changsha 410006, China

²College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China

Corresponding author: Kun Xie (xiekun@hnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572184, Grant 61472130, and Grant 61472131; in part by the Hunan Provincial Natural Science Foundation of China under Grant 2017JJ1010; in part by the Open Project Funding of CAS Key Laboratory of Network Data Science and Technology (Institute of Computing Technology, Chinese Academy of Sciences) under Grant CASNDST201704; and in part by the Outstanding Graduate Student Innovation Fund Program of Collaborative Innovation Center of High Performance Computing.

ABSTRACT Tensor completion is a higher way analog of matrix completion, which has proven to be a powerful tool for data analysis. In this paper, we formulate the missing data recovery problem of a three-way tensor as a tensor completion problem. We propose a novel tensor completion method based on tensor-CUR decomposition to estimate the missing data from limited samples. Computational experiments demonstrate that the proposed method yields a superior performance over other existing approaches.

INDEX TERMS Data recovery, tensor completion, tensor-CUR decomposition.

I. INTRODUCTION

Missing data is a commonly encountered and challenging issue which occurs in a wide range of applications [1]–[6]. To estimate the missing data, the traditional methods [7]–[9] tend to produce erroneous estimates when the ratio of missing data is high and mostly do not exploit the benefit of the hidden structure in the data.

Recently, the compressive sensing (CS) theory has been used to provide a new paradigm for the problem of missing data recovery [1], [2]. However, it was originally designed to recover the sparse vector. Matrix completion [10] has emerged due to the rapid progress of sparse representation. Matrix completion theory can accurately reconstruct a low-rank matrix from a relatively small number of entries in the matrix, which provides a new process for missing data recovery. These approaches present good performance when the data missing ratio is low. However, their performance suffers when the missing ratio is large [11], [12].

Tensor completion is a higher-way analogue of matrix completion, which can recover a low-rank tensor with limited samples. It can exploit multiple features of the data to overcome the shortcomings of the matrix-based methods, and has been successfully applied to various application fields like image analysis [13], [14], recommender systems [15], [16], wireless spectrum map construction [17], seismology signal processing [18], and computer vision [19]. Based on these

successful usage of tensor pattern, in this paper, we focus on the problem of missing data recovery of a 3-way tensor, and formulate the missing data recovery problem as a tensor completion problem.

Several tensor completion algorithms have been proposed for data recovery [20]–[22], in which the models used mainly include the CANDECOMP/PARAFAC (CP) model and the TUCKER model. CP approach requires the tensor rank (cp rank) to act as an important parameter to decompose the tensor. However, determining the rank of an arbitrary tensor is quite difficult and has been shown to be NP complete [23]. Moreover, the best rank-K approximation to a tensor may not always exist in the CP approach. Tucker decomposition is an unsupervised multiway data analysis method, and it can be considered as a high order principle component analysis. Tucker decomposition is different from CP rank because it requires n-rank. However, the suitable choices of the truncation values of n-rank are not likely to be known a priori [24].

Zhang and Aeron [25] proposed a tensor singular value decomposition based method (tensor-SVD) to estimate the missing value of the tensor. This method uses a t -product defining the multiplication action. The advantage of tensor-SVD is that the resulting algebra and analysis is very close to that of matrix algebra and analysis [25]. However, the tensor-SVD algorithm is computationally expensive, since it requires computing the SVD decomposition

of the approximate matrix at each iteration of the optimization.

To overcome the shortcomings of the existing tensor methods, in this paper, we propose a new tensor completion algorithm based on tensor-CUR decomposition. Because of the success of t -product in tensor-SVD, we also adopted it to define the multiplication action in our proposed algorithm. Unlike the tensor-SVD algorithm, our proposed algorithm extend matrix-CUR decomposition to tensor, which computes the low rank approximation of a given tensor by using the actual rows and columns of the tensor. It only needs to solve a standard regression problem and therefore, it is computationally efficient. Moreover, by using the actual rows and columns of the tensor, our proposed algorithm can efficiently compute the low rank approximation of a given tensor without having to know the rank a priori. Our goal is to exploit the tensor-CUR approximation algorithm to quickly estimate the missing data of original data tensor from its partial observations.

Our contributions can be summarized as follows:

- To overcome the shortcomings of the existing tensor decomposition models, we propose a tensor-CUR decomposition model, which computes the low rank approximation of a given tensor by using the actual rows and columns of the tensor.
- To design an efficient and accurate missing data recovery algorithm, we propose a new tensor completion algorithm based on tensor-CUR decomposition.
- We provide a theoretical guarantee of how many samples are required for an exact recovery.
- To evaluate the performance of our proposed algorithms, we have performed simulations based on real data. Compared to the standard tensor completion methods, the advantage of the proposed algorithm is its computational efficiency.

The rest of this paper is organized as follows: In Section II, we briefly introduce some essential notations and basic definitions used in this paper. The related work is introduced in Sections III. In Section IV, we formulate the missing data recovery problem. We present our proposed tensor-CUR based algorithm in Section V. In Section VI, we give our theoretical analysis. In Section VII, we give our empirical results. Section VIII concludes this paper.

II. NOTATION AND BASIC DEFINITIONS

In this section, we briefly introduce some notations and basic definitions used throughout this paper.

A. NOTATION

We denote scalars with lower-case normal letters (e.g., a), vectors with bold lower-case letters (e.g., \mathbf{a}). Matrices are denoted by boldface capital letters (e.g., \mathbf{A}). Columns of a matrix are denoted by boldface lower letters with a subscript (e.g., a_i). Entries of a matrix or a tensor are denoted by lower-case letters with subscripts (e.g., a_{ij} meaning the entry with row index i and column index j in a matrix). We frequently

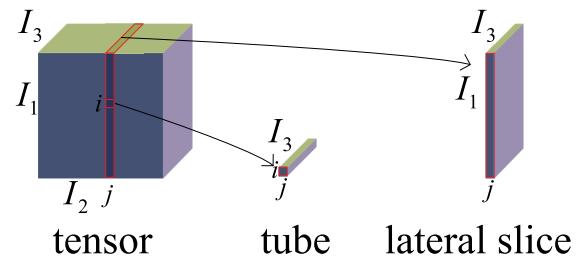


FIGURE 1. General 3-way tensor, tensor tube and tensor lateral slice.

use i, j to denote the meaning of indices and with some abuse of notation and we will use I, J to denote the index upper bounds. High-order tensors are denoted with calligraphic uppercase letters (e.g., \mathcal{A}), we further denote sets by Gothic letters (e.g., Ω). $\mathcal{A}(i, j, :)$ will denote a tensor tube which is a column vector obtained by holding i, j fixed in a tensor. $\mathcal{A}^{(k)}$ denotes the k -th frontal face. $\mathcal{A}(:, j, :)$ denotes the j -th lateral slice and $\mathcal{A}(i, :, :, :)$ denotes the i -th horizontal slice. For any third order tensor \mathcal{A} , $\hat{\mathcal{A}} = FFT(\mathcal{A}, [], 3)$ denotes the third order tensor of the same size obtained by taking the Fourier transform along the third dimension. In the same fashion, one can also compute \mathcal{A} from $\hat{\mathcal{A}}$ via $IFFT(\hat{\mathcal{A}}, [], 3)$ using the inverse FFT.

We provide details for the basic definitions used throughout this paper on the next sub-section.

B. BASIC DEFINITIONS

Definition 1 (Discrete Fourier Transform): The Discrete Fourier Transform (DFT) of a vector $x \in \mathbb{R}^n$ is a matrix-vector product

$$y = F_n x, \quad (1)$$

where the DFT matrix $F_n = (f_{kj}) \in \mathbb{R}^{n \times n}$ is defined by

$$f_{kj} = \omega_n^{(k-1)(j-1)} \quad (2)$$

$$\text{with } \omega_n = \exp\left(\frac{-2\pi i}{n}\right) = \cos\left(\frac{2\pi}{n}\right) - i \cdot \sin\left(\frac{2\pi}{n}\right).$$

Definition 2 (Kronecker product [26]): The Kronecker product of matrices A and B is denoted as matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & & a_{2n}B \\ \vdots & & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \quad (3)$$

The Kronecker product creates many copies of matrix B and scales each one by the corresponding entry of A .

Definition 3 (Tensor [25]): A tensor, also known as N^{th} -order or N -way tensor, denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, where N is the order of \mathcal{A} , also called way. The element of \mathcal{A} is denoted by $a_{i_1, i_2, \dots, i_N, i_N} \in \{1, 2, \dots, I_n\}$, $1 \leq n \leq N$. Fig.1 shows a general 3-way tensor, tensor tube and tensor lateral slice.

Definition 4 (Inner Product [27]): The Inner product of two 3-way tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ and $\mathcal{B} \in R^{I_1 \times I_2 \times I_3}$ is the sum of the products of their entries, which is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \sum_{i_3} a_{i_1 i_2 i_3} b_{i_1 i_2 i_3}. \quad (4)$$

Therefore, the norm of a tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ is defined as $\|\mathcal{A}\|^2 = \langle \mathcal{A}, \mathcal{A} \rangle$. And the corresponding (Frobenius-) norm is $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \frac{1}{\sqrt{I_3}} \|\hat{\mathcal{A}}\|_F$.

Definition 5 (Tensor Transpose [25]): The transpose of a 3-way tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ is the $I_1 \times I_2 \times I_3$ tensor \mathcal{A}^T obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slices 2 through I_3 .

Definition 6 (Circulant Matrices [28]): Given a vector $v = [v_0 \ v_1 \ v_2 \ v_3]^T$, then

$$circ(v) = \begin{bmatrix} v_0 & v_3 & v_2 & v_1 \\ v_1 & v_0 & v_3 & v_2 \\ v_2 & v_1 & v_0 & v_3 \\ v_3 & v_2 & v_1 & v_0 \end{bmatrix} \quad (5)$$

is a circulant matrix. We adopt the convention that $circ(v)$ refers to the circulant matrix obtained with the vector v as the first column.

Kilmer *et al.* [28] showed that circulant matrices can be diagonalized with the normalized Discrete Fourier Transform (DFT) matrix. In particular, if v is $n \times 1$ and F_n is the $n \times n$ DFT matrix, then $\mathcal{F}_n circ(v) \mathcal{F}_n^*$ is diagonal, where \mathcal{F}_n^* is its conjugate transpose. Kilmer *et al.* [28] also concluded that the diagonal of $\mathcal{F}_n circ(v) \mathcal{F}_n^* = fft(v)$, where $fft(v)$ denotes the Fast Fourier Transform of v .

Definition 7 (Block Circulant Matrix [28]): If $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ with I_3 frontal faces then

$$circ(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} & \mathcal{A}^{(I_3)} & \mathcal{A}^{(I_3-1)} & \dots & \mathcal{A}^{(2)} \\ \mathcal{A}^{(2)} & \mathcal{A}^{(1)} & \mathcal{A}^{(I_3)} & \dots & \mathcal{A}^{(3)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathcal{A}^{(I_3)} & \mathcal{A}^{(I_3-1)} & \ddots & \mathcal{A}^{(2)} & \mathcal{A}^{(1)} \end{bmatrix} \quad (6)$$

is a block circulant matrix of size $I_1 I_3 \times I_2 I_3$.

Kilmer *et al.* [28] also showed that block-circulant matrices can be block-diagonalized. Suppose $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ and $F_3 \in R^{I_3 \times I_3}$ is the DFT matrix. Then

$$(F_{I_3} \times I_1) \times circ(\mathcal{A}) \times (F_{I_3}^* \times I_2) = \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & & \\ & \hat{\mathcal{A}}^{(2)} & & & \\ & & \ddots & & \\ & & & \ddots & \hat{\mathcal{A}}^{(I_3)} \end{bmatrix}, \quad (7)$$

where $\hat{\mathcal{A}}$ is computed by applying FFTs along each tube $\mathcal{A}(i, j, :)$ of \mathcal{A} and $\hat{\mathcal{A}}^{(i)}$ are the frontal faces of the tensor $\hat{\mathcal{A}}$.

Definition 8 (Tensor Nuclear Norm [25]): The tensor nuclear norm of a 3-way tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ is denoted as $\|\mathcal{A}\|_{TNN}$, is the sum of singular values of all the frontal faces of $\hat{\mathcal{A}}$. In fact,

$$\|\mathcal{A}\|_{TNN} = \|\bar{\mathcal{A}}\|_*, \quad (8)$$

where $\bar{\mathcal{A}}$ denote the block-diagonal matrix of the tensor \mathcal{A}^T in the fourier domain

$$\bar{\mathcal{A}} = \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(I_3)} \end{bmatrix} \in R^{(I_1 I_3) \times (I_2 I_3)}. \quad (9)$$

Then $\|\bar{\mathcal{A}}\|_* = \sum_{i=1}^{I_3} \|\hat{\mathcal{A}}^{(i)}\|_* = \sum_{i=1}^{I_3} \sum_{j=1}^{r_i} \sigma_j(\hat{\mathcal{A}}^{(i)})$, where $r_i = rank(\hat{\mathcal{A}}^{(i)})$ is the rank of $\hat{\mathcal{A}}^{(i)}$, and $\sigma_j(\hat{\mathcal{A}}^{(i)})$ is the j^{th} singular value of $\hat{\mathcal{A}}^{(i)}$.

Definition 9 (T-Product [25]): The T-product $\mathcal{A} * \mathcal{B}$ of tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ and $\mathcal{B} \in R^{I_2 \times I_4 \times I_3}$ is an $I_1 \times I_4 \times I_3$ tensor whose $(i, j)^{th}$ tube $\mathcal{C}(i, j, :)$ is given by

$$\mathcal{C}(i, j, :) = \sum_{k=1}^{I_2} \mathcal{A}(i, k, :) * \mathcal{B}(k, j, :), \quad (10)$$

where * denotes the circular convolution between two tubes of the same size. We anchor the *MatVec* command to the frontal faces of the tensor. *MatVec*(\mathcal{A}) takes an $I_1 \times I_2 \times I_3$ tensor and returns a block $I_1 I_2 \times I_3$ matrix, whereas the *fold* command undoes this operation

$$MatVec(\mathcal{A}) = \begin{bmatrix} \mathcal{A}^{(1)} \\ \mathcal{A}^{(2)} \\ \vdots \\ \mathcal{A}^{(I_3)} \end{bmatrix}, \quad fold(MatVec(\mathcal{A})) = \mathcal{A}. \quad (11)$$

Then the T-product $\mathcal{A} * \mathcal{B}$ is the $I_1 \times I_4 \times I_3$ tensor

$$\mathcal{A} * \mathcal{B} = fold(circ(\mathcal{A}).MatVec(\mathcal{B})) \quad (12)$$

Zhang and Aeron [25] showed that for any tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ and $\mathcal{B} \in R^{I_2 \times I_4 \times I_3}$, then $\mathcal{A} * \mathcal{B} = \mathcal{C} \Leftrightarrow \bar{\mathcal{A}} \bar{\mathcal{B}} = \bar{\mathcal{C}}$. We are able to transform the t-product into its equivalent form in Fourier domain. On the other hand, we can also transform an operator in Fourier domain back to the original domain as needed.

Definition 10 (Identity Tensor [25]): The identity tensor $\mathcal{I} \in R^{n_1 \times n_1 \times n_3}$ is defined to be a tensor whose first frontal slice $\mathcal{I}^{(1)}$ is the $n_1 \times n_1$ identity matrix and all other frontal slices $\mathcal{I}^{(i)}$; $i = 2, \dots, n_3$ are zero.

Definition 11 (Orthogonal Tensor [25]): A tensor $\mathcal{A} \in R^{n_1 \times n_1 \times n_3}$ is orthogonal if it satisfies

$$\mathcal{A}^T * \mathcal{A} = \mathcal{A} * \mathcal{A}^T = \mathcal{I} \quad (13)$$

Definition 12 (Inverse of Tensor [25]): The inverse of a tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ is written as \mathcal{A}^{-1} satisfying

$$\mathcal{A}^{-1} * \mathcal{A} = \mathcal{A} * \mathcal{A}^{-1} = \mathcal{I} \quad (14)$$

III. RELATED WORK

In this section, we review the related work on tensor completion and then show the differences between our work and the existing work. We mainly consider three types of tensor completion methods, which are based on the CP format, the tucker decomposition and the tensor singular value decomposition respectively. Below we provide details for each of these methods.

A. TENSOR COMPLETION BASED ON CP DECOMPOSITION

CANDECOMP/PARAFAC (CP) decomposition [29], [30] is a powerful tool for data analysis. Evirm et al. proposed a CP weighted optimization algorithm to estimate the missing value of tensor [21]. The CP decomposition of a tensor $\mathcal{A} \in R^{I_1 \times I_2 \times \dots \times I_N}$ of order N is given by

$$\mathcal{A} = \sum_{l=1}^L a_l^{(1)} \circ a_l^{(2)} \circ \dots \circ a_l^{(N)}, \quad a_l^{(n)} \in R^{I_n}, \quad (15)$$

where \circ denotes the outer product of the vector. It consists of decomposition of a given tensor as a sum of the lowest possible number of rank-one tensors. The CP rank is defined by the minimum number of rank-one terms in CP decomposition. Fig.2 show the CP decomposition of a 3-way tensor.

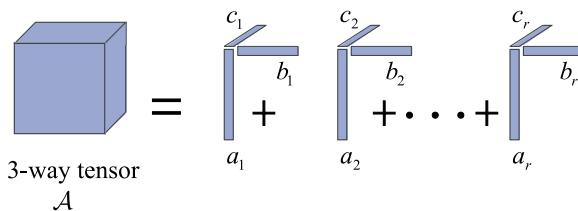


FIGURE 2. The CP decomposition of a 3-way tensor.

Suppose we sample a 3-way tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ at the set of indices in a set Ω . Let P_Ω denotes the sampling operator $P_\Omega : R^{I_1 \times I_2 \times I_3} \rightarrow R^{I_1 \times I_2 \times I_3}$, which is defined by

$$P_\Omega(\mathcal{A})_{ijk} = \begin{cases} \mathcal{A}_{ijk} & (i, j, k) \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

Jain and Oh [30] tried to complete the tensor by solving the following optimization problem

$$\underset{\hat{\mathcal{A}}, \text{rank}(\hat{\mathcal{A}})=r}{\text{minimize}} \left\| P_\Omega \left(\mathcal{A} - \sum_{l=1}^r \delta_l \left(a_l^{(1)} \circ a_l^{(2)} \circ a_l^{(3)} \right) \right) \right\|_F^2 \quad (17)$$

They showed that under certain standard assumptions, the proposed method can recover a three-mode $n \times n \times n$ dimensional rank- r tensor exactly from $O(n^{3/2}r^5 \log^4 n)$ randomly sampled entries.

For CP approach, determining the rank of an arbitrary tensor is quite difficult and has been shown to be NP-complete [23]. It is often computationally difficult to determine the CP rank or the best low rank CP approximation of a practical tensor data beforehand. In contrast to

CP approach, our proposed algorithm extend matrix-CUR decomposition to tensor, which can efficiently compute the low rank approximation of a given tensor by using the actual rows and columns of the tensor.

B. TENSOR COMPLETION BASED ON TUCKER DECOMPOSITION

Tucker decomposition [31] is an unsupervised multiway data analysis method, and it can be considered as a high order principle component analysis. Tan et al. [22] proposed a Tucker decomposition based method to estimate the missing value of the tensor. The basic idea of the Tucker decomposition is to decompose a tensor into a small core tensor multiplied by a factor matrix along each mode. The Tucker decomposition of an N^{th} order tensor $\mathcal{A} \in R^{I_1 \times I_2 \times \dots \times I_N}$ can be written as

$$\mathcal{A} = \zeta \times_1 A^{(1)} \times_2 A^{(2)} \dots \times_N A^{(N)}, \quad (18)$$

where $\zeta \in R^{R_1 \times R_2 \times \dots \times R_N}$ is the core tensor, and $A^{(n)} = [a_1^{(n)} \dots a_{R_n}^{(n)}] \in R^{I_n \times R_n}$ denotes the factor matrix along the n^{th} mode. Fig.3 illustrates the Tucker-3 decomposition of a 3-way tensor.

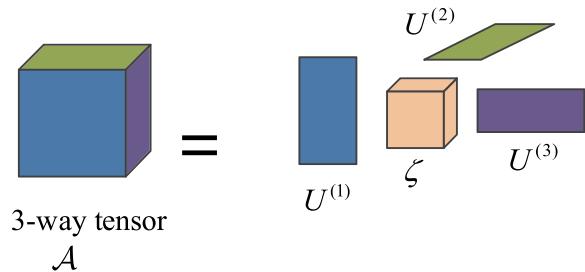


FIGURE 3. The Tucker3 Decomposition of a 3-way tensor.

Liu et al. [19] proposed a tensor completion algorithm based on minimizing tensor n -rank in Tucker decomposition format. It uses the matrix nuclear norm instead of matrix rank, and try to solve the convex problem as follows

$$\min_{\mathcal{X}} \sum_{i=1}^n \alpha_i \|X_{(i)}\|_* \quad \text{subject to } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{A}), \quad (19)$$

where $X_{(i)}$ is the mode- n matricization of \mathcal{X} and $\alpha_{(i)}$ are prespecified constants satisfying $\alpha_{(i)} \geq 0$, $\sum_{i=1}^n \alpha_{(i)} = 1$.

Unlike the optimal dimensionality reduction in matrix PCA which can be obtained by truncating the SVD, there is no trivial multi-linear counterpart to dimensionality reduction for Tucker type. Alternatively, suitable choices of the truncation values of n-rank are not likely to be known a priori [24]. As a contrast, our proposed algorithm use the actual rows and columns of the tensor to efficiently compute the low rank approximation of a given tensor, without having to know the rank a priori.

C. TENSOR COMPLETION BASED ON TENSOR SINGULAR VALUE DECOMPOSITION

Zhang and Aeron [25] considered sampling and recovery for a 3-way tensor using the tensor singular value decomposition (tensor-SVD), in which it uses a t -product defining the multiplication action. The tensor-SVD of a tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ of order 3 is given by

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T, \quad (20)$$

where \mathcal{U} and \mathcal{V} are orthogonal tensors of size $I_1 \times I_1 \times I_3$ and $I_2 \times I_2 \times I_3$ respectively. \mathcal{S} is a rectangular f -diagonal tensor [25] of size $I_1 \times I_2 \times I_3$, and the entries in \mathcal{S} are called the singular values of \mathcal{A} . Tensors \mathcal{U} , \mathcal{S} and \mathcal{V} are derived from individual matrix-SVD in Fourier space and $*$ denotes the t -product. Fig.4 show the tensor singular value decomposition of a 3-way tensor.

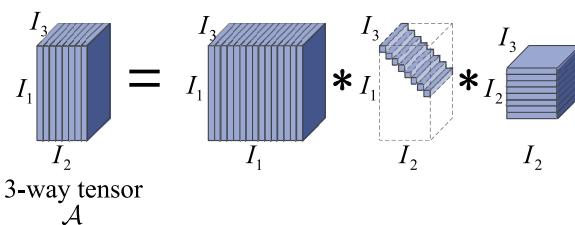


FIGURE 4. The tensor-SVD of an $I_1 \times I_2 \times I_3$ tensor.

The tensor tubal rank is used in tensor-SVD, also referred to as tensor rank. Zhang et al. tried to complete the tensor by solving the following convex optimization problem

$$\begin{aligned} & \min_{\mathcal{X}} \|\mathcal{X}\|_{TNN} \\ & \text{subject to } P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{A}), \end{aligned} \quad (21)$$

where the tensor nuclear norm is taken as the convex relaxation of tensor tubal rank, $\|\cdot\|_{TNN}$ is the tensor nuclear norm. Zhang et al. showed that one can perfectly recover a tensor of size $I_1 \times I_2 \times I_3$ with rank r under tensor-SVD as long as $O(rI_1I_3\log((I_1+I_2)I_3))$ samples are observed.

The success of tensor-SVD is based on the usage of t -product, which ensures that the result of algebra and analysis is very close to that of matrix algebra and analysis [25]. Because of the success of the t -product in our proposed algorithm, we chose to use it to define the multiplication action.

However, the tensor-SVD algorithm is computationally expensive, especially for a large tensor, because it requires computing the SVD decomposition of the approximate matrix at each iteration of the optimization. To overcome this shortcoming of tensor-SVD based algorithm, our proposed algorithm extends the recently developed matrix-CUR decomposition to tensor. It computes the low rank approximation of a given tensor by using the actual rows and columns of the tensor. It only needs to solve a standard regression problem and therefore it is computationally efficient.

IV. PROBLEM FORMULATION

In this section, we formulate the missing data recovery problem as a tensor completion problem. In order to estimate the missing data, we propose a tensor-CUR decomposition model. Since our proposed tensor-CUR decomposition is a 3-way analogue of matrix-CUR decomposition, below we first introduce CUR approximation.

Definition 13 (Matrix CUR Approximation (M-CUR)) [32]–[34]: Let A be an $m \times n$ matrix. In addition, let C be an $m \times d$ matrix whose columns consist of a small number of d columns of the matrix A , let R be an $l \times n$ matrix whose rows consist of a small number of l rows of the original matrix A , and let U be a $d \times l$ matrix. Then Y is a column-row-based low-rank approximation, or a CUR approximation to A , if it may be explicitly written as

$$Y = C \times U \times R. \quad (22)$$

This decomposition provides a low rank approximation $Y = C \times U \times R$ when the number of selected rows and columns is lower than the $\min(m, n)$.

CUR matrix decomposition can efficiently compute the low rank approximation for a given matrix by using the actual rows and columns of the matrix. It has been used as a very useful tool for handling large matrices. Compared to other low rank approximation algorithms, CUR matrix decomposition only needs to solve a standard regression problem and therefore, can be solved efficiently. It does not have to compute the SVD decomposition of the approximate matrix at each iteration of the optimization.

Our proposed tensor-CUR decomposition is a 3-way analogue of matrix-CUR decomposition. It inherits the advantage of matrix-CUR decomposition and utilize it into our missing data recovery problem to obtain a higher efficiency. We give the definition of the tensor CUR decomposition below.

Definition 14 (Tensor CUR Decomposition (T-CUR)): For a 3-way tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$, the tensor-CUR decomposition is given by

$$\mathcal{A} = \mathcal{C} * \mathcal{U} * \mathcal{R}, \quad (23)$$

where \mathcal{C} , \mathcal{U} and \mathcal{R} are tensors of size $I_1 \times I_1 \times I_3$, $I_1 \times I_2 \times I_3$ and $I_2 \times I_2 \times I_3$ respectively.

The missing data recovery problem can be formulated as a tensor completion problem with the goal of finding its missing entries through the following optimization

$$\begin{aligned} & \min_{\mathcal{U} \in R^{I_1 \times I_2 \times I_3}} \frac{1}{2} \|P_{\Omega}(\mathcal{Y} - \mathcal{C} * \mathcal{U} * \mathcal{R})\|_F^2 \\ & \text{subject to } P_{\Omega}(\mathcal{Y}) = P_{\Omega}(\mathcal{A}) \end{aligned} \quad (24)$$

V. PROPOSED ALGORITHM

In this section, we propose a novel tensor completion algorithm to accurately recover the missing data. Our goal is to develop a tensor-CUR approximation algorithm to estimate the missing data.

A. TENSOR COMPLETION WITH TUBAL SAMPLING

Tensor completion via uniform entry-wise sampling is well-studied. Its other perspective is by random tubal sampling as considered in [25]. In this subsection, we formally define the tensor completion problem with tubal sampling. Fig.5 illustrate the operation of tubal sampling.

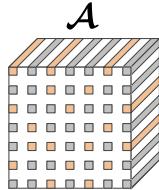


FIGURE 5. Tubal sampling.

For a given 3-way tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$, suppose there are m tubes of \mathcal{A} , sampled according to the Bernoulli model. It means that each tensor tube of \mathcal{A} is sampled along the third dimension with probability p independent of others. The task of tensor completion problem is to recover \mathcal{A} from the observed entries. Noting that $\mathcal{Y} = \mathcal{C} * \mathcal{U} * \mathcal{R} \Leftrightarrow \bar{\mathcal{Y}} = \bar{\mathcal{C}} \bar{\mathcal{U}} \bar{\mathcal{R}}$ [25], makes it much simpler to establish that solving this optimization problem (24) is equivalent to solving the following problem in the Fourier domain under tubal sampling [25]:

$$\min_{\bar{\mathcal{U}} \in R^{(I_1 I_3) \times (I_2 I_3)}} \frac{1}{2} \|P_\Omega(\bar{\mathcal{Y}} - \bar{\mathcal{C}} \times \bar{\mathcal{U}} \times \bar{\mathcal{R}})\|_F^2 \\ \text{subject to } \mathcal{F}_{I_3} P_\Omega \mathcal{F}_{I_3}^{-1}(\bar{\mathcal{Y}}) = \mathcal{F}_{I_3} P_\Omega \mathcal{F}_{I_3}^{-1}(\bar{\mathcal{A}}), \quad (25)$$

where \mathcal{F}_{I_3} is a mapping which maps a third order tensor \mathcal{Z} to $\bar{\mathcal{Z}}$ along the third dimension of tensors, $\mathcal{F}_{I_3}^{-1}$ is its inverse transform. In this case, one is given observations of \mathcal{A} at indices $(i, j) \in \Omega$, $i \in 1, 2, \dots, I_1$, $j \in 1, 2, \dots, I_2$. Noting that:

$$P_\Omega(\bar{\mathcal{Y}} - \bar{\mathcal{C}} \times \bar{\mathcal{U}} \times \bar{\mathcal{R}}) \\ = P_\Omega \left(\begin{bmatrix} \hat{\mathcal{Y}}^{(1)} - \hat{\mathcal{C}}^{(1)} \times \hat{\mathcal{U}}^{(1)} \times \hat{\mathcal{R}}^{(1)} \\ \vdots \\ \hat{\mathcal{Y}}^{(I_3)} - \hat{\mathcal{C}}^{(I_3)} \times \hat{\mathcal{U}}^{(I_3)} \times \hat{\mathcal{R}}^{(I_3)} \end{bmatrix} \right) \quad (26)$$

Then, solving this optimization problem in (25) is equivalent to solving the following I_3 matrix completion problem under linear constraint [25]:

$$\min_{\hat{\mathcal{U}}^{(k)} \in R^{(I_1 I_3) \times (I_2 I_3)}} \frac{1}{2} \|P_\Omega(\hat{\mathcal{Y}}^{(k)} - \hat{\mathcal{C}}^{(k)} \times \hat{\mathcal{U}}^{(k)} \times \hat{\mathcal{R}}^{(k)})\|_F^2 \\ \text{subject to } \mathcal{F}_{I_3} P_\Omega \mathcal{F}_{I_3}^{-1}(\hat{\mathcal{Y}}^{(k)}) = \mathcal{F}_{I_3} P_\Omega \mathcal{F}_{I_3}^{-1}(\hat{\mathcal{A}}^{(k)}). \quad (27)$$

Therefore tensor completion problem with tubal sampling is essentially the matrix completion from random samplings of each face in the Fourier domain. We can directly use some result of matrix-CUR approximation algorithm. Fig.6 illustrate the transformation process.

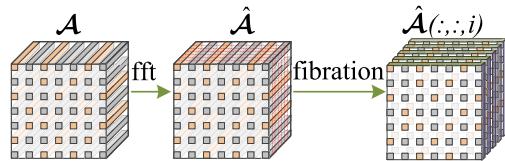


FIGURE 6. Tensor completion with tubal sampling.

B. TENSOR-CUR COMPLETION

In this subsection, we provide details of our tensor-CUR completion algorithm. Algorithm 1 gives our T-CUR algorithm and Fig.7 illustrates the process of our T-CUR algorithm.

Algorithm 1 Tensor-CUR Approximation From Partially Observed Entries (T-CUR)

Require: Observation data $P_\Omega(\mathcal{A}) \in R^{I_1 \times I_2 \times I_3}$, vector of sample columns/rows number $d/l \in R^{1 \times I_3}$

- 1: $\hat{\mathcal{A}} = FFT(P_\Omega(\mathcal{A}), [], 3);$
- 2: **for** $i \leftarrow 1, \dots, I_3$ **do**
- 3: $[\hat{\mathcal{C}}^{(i)}, \hat{\mathcal{U}}^{(i)}, \hat{\mathcal{R}}^{(i)}] = M\text{-CUR}\left(P_\Omega(\hat{\mathcal{A}}^{(i)}), d_i, l_i\right)$
- 4: **end for**
- 5: $\mathcal{C} = IFFT(\hat{\mathcal{C}}, [], 3); \mathcal{U} = IFFT(\hat{\mathcal{U}}, [], 3); \mathcal{R} = IFFT(\hat{\mathcal{R}}, [], 3)$
- 6: Return a tensor cur approximation of \mathcal{A} : $\mathcal{Y} = \mathcal{C} * \mathcal{U} * \mathcal{R} \in R^{I_1 \times I_2 \times I_3}$

On line 1, as initial step, we take the Fourier transform along the third dimension of the observation data to obtain a third order tensor of the same size (see Fig.7(a) to Fig.7(b)). On lines 2-4, we solve the M-CUR approximation of each frontal face $\hat{\mathcal{A}}^{(i)}$ by executing Algorithm 2, where both d_i and l_i are parameters which are determined following the Theorem 3. As shown, only the observed entries are utilized. After getting three factor tensor $\hat{\mathcal{C}}$, $\hat{\mathcal{U}}$ and $\hat{\mathcal{R}}$ in the Fourier domain (see Fig.7(d) and Fig.7(e)), on line 5, we use the inverse Fourier to transform these factor tensor into three factor tensor \mathcal{C} , \mathcal{U} and \mathcal{R} . This will then be used for tensor CUR approximation in the original domain (see Fig.7(b) to Fig.7(a)). Lastly, \mathcal{Y} is the final recovered tensor CUR approximation of \mathcal{A} , which is calculated by $\mathcal{Y} = \mathcal{C} * \mathcal{U} * \mathcal{R}$.

One limitation of the existing algorithms for CUR matrix decomposition is that they require an access to the full matrix to effectively compute the low rank approximation [35]. Mackey *et al.* [32] alleviated this limitation by developing a CUR decomposition algorithm for partially observed matrices.

In particular, for $I_1 \times I_2$ matrix $\hat{\mathcal{A}}^{(i)}$, let Ω represent the set of indices of observed entries. Let $\tilde{\mathcal{C}}^{(i)}$ be an $I_1 \times d_i$ matrix whose columns sampled uniformly at random from the columns of $P_\Omega(\hat{\mathcal{A}}^{(i)})$, and let $\tilde{\mathcal{R}}^{(i)}$ be an $I_2 \times l_i$ matrix whose rows are sampled uniformly at random from the rows of $P_\Omega(\hat{\mathcal{A}}^{(i)})$. The proposed algorithm computes the low rank approximation of $\tilde{\mathcal{C}}^{(i)}$ and $\tilde{\mathcal{R}}^{(i)}$ by using the matrix completion technique. Let r_i be the target rank for approximation, with

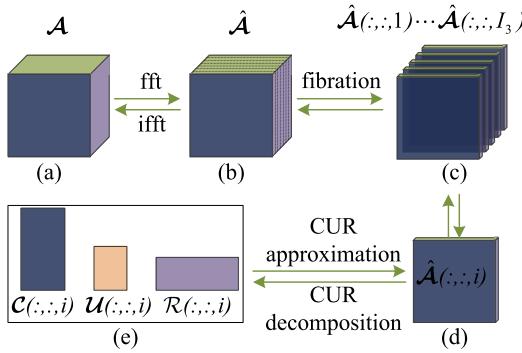


FIGURE 7. The Tensor-CUR approximation of a 3-way tensor.

$r_i \leqslant \min(d_i, l_i)$. Let $\hat{\mathcal{C}}^{(i)}$ and $\hat{\mathcal{R}}^{(i)}$ be the low-rank r_i approximations of $\tilde{\mathcal{C}}^{(i)}$ and $\tilde{\mathcal{R}}^{(i)}$, respectively. Then, the goal is to estimate a low rank approximation of matrix $\hat{\mathcal{A}}^{(i)}$ using $\hat{\mathcal{C}}^{(i)}$, $\hat{\mathcal{R}}^{(i)}$ and the randomly sampled entries in Ω . So far, it needs to solve the following optimization:

$$\begin{aligned} \min_{\hat{\mathcal{U}}^{(i)} \in R^{r_i \times r_i}} \quad & \frac{1}{2} \|P_\Omega(\hat{\mathcal{Y}}^{(i)} - \hat{\mathcal{C}}^{(i)} \hat{\mathcal{U}}^{(i)} \hat{\mathcal{R}}^{(i)})\|_F^2 \\ \text{subject to } \quad & P_\Omega(\hat{\mathcal{Y}}^{(i)}) = P_\Omega(\hat{\mathcal{A}}^{(i)}). \end{aligned} \quad (28)$$

Note that (28) is a standard regression problem [35], which can be solved efficiently using the standard regression method. Algorithm 2 summarize the general framework that we refer to as M-CUR.

Algorithm 2 Matrix-CUR Approximation From Partially Observed Entries (M-CUR)

Require: $P_\Omega(\hat{\mathcal{A}}^{(i)})$, d_i , l_i .

- 1: $P_\Omega(\tilde{\mathcal{C}}^{(i)})$ = randomly sample d_i columns of $P_\Omega(\hat{\mathcal{A}}^{(i)})$.
- 2: $P_\Omega(\tilde{\mathcal{R}}^{(i)})$ = randomly sample l_i rows of $P_\Omega(\hat{\mathcal{A}}^{(i)})$.
- 3: do
 $\hat{\mathcal{C}}^{(i)} = MC\text{-BASE}(P_\Omega(\tilde{\mathcal{C}}^{(i)}))$
 $\hat{\mathcal{R}}^{(i)} = MC\text{-BASE}(P_\Omega(\tilde{\mathcal{R}}^{(i)}))$
- 4: end do
- 5: Let $\hat{\mathcal{W}}^{(i)}$ be the $l_i \times d_i$ matrix formed from the intersection of the sampled rows and columns.
- 6: Return $\hat{\mathcal{C}}^{(i)}$, $\hat{\mathcal{W}}^{(i)\dagger}$, $\hat{\mathcal{R}}^{(i)}$, and then get a Matrix-CUR approximation of $\hat{\mathcal{A}}^{(i)}$: $\hat{\mathcal{Y}}^{(i)} = (\hat{\mathcal{C}}^{(i)} \hat{\mathcal{W}}^{(i)\dagger} \hat{\mathcal{R}}^{(i)})$.

On line 3, *MC-BASE* denote a standard base matrix factorization algorithm which output the low-rank approximations of its input. We use the state-of-the-art singular value thresholding (SVT) approach [36] as our base MC algorithm, which can automatically specify the rank of the unknown matrix by using a sample method. For more details see [36]. After we got the two low-rank approximations $\hat{\mathcal{C}}^{(i)}$ and $\hat{\mathcal{R}}^{(i)}$, on line 6, we calculate the matrix-CUR approximation of $\hat{\mathcal{A}}^{(i)}$ as $\hat{\mathcal{Y}}^{(i)} = (\hat{\mathcal{C}}^{(i)} \hat{\mathcal{W}}^{(i)\dagger} \hat{\mathcal{R}}^{(i)})$, where \dagger is the pseudo-inverse [32].

C. COMPLEXITY ANALYSIS

In this subsection, we compare computational complexity for T-SVD and T-CUR.

As the main steps of T-SVD are matrix-SVD of each frontal face, the main complexity of T-SVD comes from computation of the matrix singular value decomposition of each frontal face. Matrix-SVD have $O(I_1 \times I_2 \times r_{\hat{\mathcal{A}}^{(i)}})$ per-iteration time complexity due to the rank- $r_{\hat{\mathcal{A}}^{(i)}}$ truncated SVD performed on each iteration, where rank- $r_{\hat{\mathcal{A}}^{(i)}}$ is the rank of the frontal face $\hat{\mathcal{A}}^{(i)}$. Then, T-SVD has a complexity time of $O(I_3 \times Iter_{max} \times I_1 \times I_2 \times \bar{r}_0)$, where $\bar{r}_0 = \max(r_{\hat{\mathcal{A}}^{(i)}})$ and $Iter_{max}$ is the value of the max iteration.

Compared to T-SVD, the main steps of T-CUR are matrix-CUR of each frontal face, as shown in the line 3 of Algorithm 1. Moreover, from Algorithm 2, we can see that the main step of matrix-CUR is on line 3, which divides the expensive task of matrix factorization into smaller subproblems. The step 3 of Algorithm 2 significantly reduces the per-iteration complexity times to $O(I_1 \times d_i \times r_{\tilde{\mathcal{C}}^{(i)}})$ for $\tilde{\mathcal{C}}^{(i)}$ and to $O(I_2 \times l_i \times r_{\tilde{\mathcal{R}}^{(i)}})$ for $\tilde{\mathcal{R}}^{(i)}$, where $r_{\tilde{\mathcal{C}}^{(i)}}$ and $r_{\tilde{\mathcal{R}}^{(i)}}$ are rank of $\tilde{\mathcal{C}}^{(i)}$ and $\tilde{\mathcal{R}}^{(i)}$, respectively. Hence, T-CUR has a similar complexity on the order of $O(I_3 \times Iter_{max} \times (I_1 \times \bar{d} + I_2 \times \bar{l}) \times \bar{r}_1)$, where $\bar{r}_1 = \max(r_{\tilde{\mathcal{C}}^{(i)}}, r_{\tilde{\mathcal{R}}^{(i)}})$, $\bar{d} = \max(d_i)$ and $\bar{l} = \max(l_i)$.

For $\bar{d}, \bar{l} < \min(I_1, I_2)$, we can see that the computational complexity of T-CUR is much lower than that of T-SVD, which demonstrates the computational efficiency of T-CUR.

VI. THEORETICAL ANALYSIS

In this section, we present our theoretical analysis. The following theorem presents a guarantee for the tensor-CUR decomposition of any real-valued tensor $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$.

Theorem 1: Suppose \mathcal{A} is an $I_1 \times I_2 \times I_3$ real-valued tensor, then \mathcal{A} can be factored as:

$$\mathcal{A} = \mathcal{C} * \mathcal{U} * \mathcal{R}, \quad (29)$$

where \mathcal{C} , \mathcal{U} and \mathcal{R} are tensors of size $I_1 \times I_1 \times I_3$, $I_1 \times I_2 \times I_3$ and $I_2 \times I_2 \times I_3$ respectively.

Proof: Firstly, we transform $circ(\mathcal{A})$ into Fourier domain.

$$\begin{aligned} & (\mathcal{F}_{I_3} \times I_{I_1}) \times circ(\mathcal{A}) \times (\mathcal{F}_{I_3}^* \times I_{I_2}) \\ &= \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(I_3)} \end{bmatrix} \end{aligned} \quad (30)$$

Secondly, we compute the M-CUR of each $\hat{\mathcal{A}}^{(i)}$ as $\hat{\mathcal{A}}^{(i)} = \hat{\mathcal{C}}^{(i)} \hat{\mathcal{U}}^{(i)} \hat{\mathcal{R}}^{(i)}$, then

$$\begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(I_3)} \end{bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} \hat{\mathcal{C}}^{(1)} & & \\ & \hat{\mathcal{C}}^{(2)} & \\ & & \ddots & \\ & & & \hat{\mathcal{C}}^{(I_3)} \end{bmatrix} \\
&\times \begin{bmatrix} \hat{\mathcal{U}}^{(1)} & & \\ & \hat{\mathcal{U}}^{(2)} & \\ & & \ddots & \\ & & & \hat{\mathcal{U}}^{(I_3)} \end{bmatrix} \\
&\times \begin{bmatrix} \hat{\mathcal{R}}^{(1)} & & \\ & \hat{\mathcal{R}}^{(2)} & \\ & & \ddots & \\ & & & \hat{\mathcal{R}}^{(I_3)} \end{bmatrix} \quad (31)
\end{aligned}$$

Since

$$\begin{aligned}
&\mathcal{F}_{I_3}^* \otimes I_{I_1} \begin{bmatrix} \hat{\mathcal{C}}^{(1)} & & \\ & \hat{\mathcal{C}}^{(2)} & \\ & & \ddots & \\ & & & \hat{\mathcal{C}}^{(I_3)} \end{bmatrix} \mathcal{F}_{I_3} \otimes I_{I_1} \\
&\mathcal{F}_{I_3}^* \otimes I_{I_1} \begin{bmatrix} \hat{\mathcal{U}}^{(1)} & & \\ & \hat{\mathcal{U}}^{(2)} & \\ & & \ddots & \\ & & & \hat{\mathcal{U}}^{(I_3)} \end{bmatrix} \mathcal{F}_{I_3} \otimes I_{I_2} \\
&\mathcal{F}_{I_3}^* \otimes I_{I_2} \begin{bmatrix} \hat{\mathcal{R}}^{(1)} & & \\ & \hat{\mathcal{R}}^{(2)} & \\ & & \ddots & \\ & & & \hat{\mathcal{R}}^{(I_3)} \end{bmatrix} \mathcal{F}_{I_3} \otimes I_{I_2}
\end{aligned}$$

are circulant matrices, we can obtain an expression for $\text{unfold}(\mathcal{A})$, by applying the appropriate matrix $\mathcal{F}_{I_3} \otimes I_{I_1}$ to the left and $\mathcal{F}_{I_3}^* \otimes I_{I_2}$ to the right of each of the block diagonal matrices in (31) to get three block circulant matrices. We define $\text{unfold}(\mathcal{C})$, $\text{unfold}(\mathcal{U})$ and $\text{unfold}(\mathcal{R})$ as the first block columns of each of the respective block-circulant matrices, and then folding up the result. This gives a decomposition of the form $\mathcal{C} * \mathcal{U} * \mathcal{R}$, and the proof is complete.

We then define two standard notions of coherence, which is used for the sufficient conditions for accurate matrix completion, and thereafter, present the theoretical guarantee for M-CUR.

Definition 15 (u_0 -Coherence [32]): Let $V \in R^{n \times r}$ contain orthonormal columns with $r \leq n$. Then, the u_0 -Coherence of V is:

$$u_0(V) = \frac{n}{r} \max_{1 \leq i \leq n} \|P_V e_i\|^2 = \frac{n}{r} \max_{1 \leq i \leq n} \|V_{(i)}\|^2,$$

where P_V is the orthogonal projection onto the column space of V , and e_i is the i^{th} column of the standard basis.

Definition 16 (u_1 -Coherence [32]): Let $L \in R^{m \times n}$ have rank r . Then, the u_1 -Coherence of L is:

$$u_1(L) = \sqrt{\frac{mn}{r}} \max_{ij} |e_i^T U_L V_L^T e_j|$$

where $U_L \in R^{m \times r}$ and $V_L \in R^{n \times r}$ are the corresponding left and right singular vectors of L .

For any $u > 0$, we call matrix L (u, r) -coherent if $\text{rank}(L) = r$, $\max(u_0(U_L), u_0(V_L)) \leq u$ and $u_1(L) \leq \sqrt{u}$. Our analysis focuses on base MC algorithms that express their recovery guarantees in terms of the (u, r) -coherence of the target low-rank matrix, where lower values of u correspond to better recovery properties.

Theorem 2 (Corollary 4 [32]): Fixing any $\varepsilon, \delta \in (0, 1)$, suppose L is the best low rank- r approximation of an $m \times n$ matrix A (this means that $\|(A - L)\|_F \leq \Delta$, where Δ is the noise level), and it is (u, r) -coherent. There are s entries of A observed, and the locations of Ω distributed uniformly. Then define an over-sampling parameter as:

$$\beta_s = \frac{s(1 - \frac{\varepsilon}{2})}{32u^2r^2(m+n)\log^2(m+n)}$$

and fix any target rate parameter $1 < \beta < \beta_s$. If

$$\begin{aligned}
l &\geq \max\left(\frac{n\beta}{\beta_s} + \sqrt{\frac{n(\beta-1)}{\beta_s}}, \text{cru} \frac{\log(n)\log(\frac{2}{\delta})}{\epsilon^2}\right) \\
d &\geq \max\left(\frac{m\beta}{\beta_s} + \sqrt{\frac{m(\beta-1)}{\beta_s}}, \text{clu}_0 \frac{\log(m)\log(\frac{4}{\delta})}{\epsilon^2}\right)
\end{aligned}$$

Then

$$\|L - \widehat{\mathcal{C}} \widehat{\mathcal{W}}^+ \widehat{\mathcal{R}}\|_F \leq (2 + 3\varepsilon)c_\epsilon \sqrt{ml + dn}\Delta$$

with the probability of at least

$$(1 - \delta)(1 - \delta - 0.2)(1 - 10\log(\bar{n})\bar{n}^{2-2\beta})$$

with c and c_ϵ as fixed positive constant and $\bar{n} = \max(m, n)$.

For more detailed proof see [32]. Suppose there are s third-dimensional tubes of $\mathcal{A} \in R^{I_1 \times I_2 \times I_3}$ sampled according to the Bernoulli model, the following theorem will present the theoretical guarantee for our T-CUR.

Theorem 3: Suppose $\mathcal{Y} \in R^{I_1 \times I_2 \times I_3}$ is the solution of algorithm 1, whose tensor-CUR is given by $\mathcal{Y} = \mathcal{C} * \mathcal{U} * \mathcal{R}$, where \mathcal{C} , \mathcal{U} and \mathcal{R} are $I_1 \times I_1 \times I_3$, $I_1 \times I_2 \times I_3$ and $I_2 \times I_2 \times I_3$ tensors, respectively. Suppose $\widehat{\mathcal{L}}^{(i)}$ is the best low rank- r_i approximation of $I_1 \times I_2$ matrix $\widehat{\mathcal{A}}^{(i)}$ (that means $\|\widehat{\mathcal{A}}^{(i)} - \widehat{\mathcal{L}}^{(i)}\|_F \leq \Delta$), and it is (u_i, r_i) -coherent. Suppose l_i and d_i are parameters of $\widehat{\mathcal{A}}^{(i)}$ consistent with Theorem 2. Let $l = \max(l_i)$ and $d = \max(d_i)$. There are s tubes of \mathcal{A} observed, and the locations of Ω distributed uniformly. Then define an Over-sampling parameter as:

$$\beta_s = \frac{s(1 - \frac{\varepsilon}{2})}{32\bar{u}^2\bar{r}^2(I_1 + I_2)\log^2(I_1 + I_2)}$$

and fix any target rate parameter at $1 < \beta < \beta_s$. If

$$l \geq \max\left(\frac{I_2\beta}{\beta_s} + \sqrt{\frac{I_2(\beta-1)}{\beta_s}}, c\bar{r} \times \bar{u} \frac{\log(I_2)\log(\frac{2}{\delta})}{\epsilon^2}\right),$$

$$d \geq \max\left(\frac{I_1\beta}{\beta_s} + \sqrt{\frac{I_1(\beta-1)}{\beta_s}}, cl\bar{u}_0 \frac{\log(I_1)\log(\frac{4}{\delta})}{\epsilon^2}\right).$$

Then

$$\|\mathcal{A} - \mathcal{Y}\|_F \leq (2 + 3\epsilon)c_\epsilon\sqrt{I_1lI_3 + dI_2I_3}\Delta$$

with probability of at least

$$(1 - \delta)(1 - \delta - 0.2)(1 - 10\log(\bar{n}))$$

with all of $(\Delta, \epsilon, \delta, c$ and $c_\epsilon)$ as shown in Theorem 2, where $\bar{n} = \max(I_1, I_2)$, $\bar{r} = \max(r_i)$, $\bar{u} = \max(u_i)$ and $\bar{u}_0 = \max((u_0)_i)$.

Proof: For every $\hat{\mathcal{A}}^{(i)}$, let

$$(\beta_s)_i = \frac{s\left(1 - \frac{\epsilon}{2}\right)}{32u_i^2r_i^2(I_1 + I_2)\log^2(I_1 + I_2)}$$

and let

$$\beta_s = \frac{s\left(1 - \frac{\epsilon}{2}\right)}{32\bar{u}^2\bar{r}^2(I_1 + I_2)\log^2(I_1 + I_2)}$$

Then we can fix any target rate parameter at $1 < \beta_i \leq \beta_s < (\beta_s)_i$, according to Theorem 2, if

$$l_i \geq \max\left(\frac{I_2\beta_i}{(\beta_s)_i} + \sqrt{\frac{I_2(\beta_i-1)}{(\beta_s)_i}}, cr_i \times u_i \frac{\log(I_2)\log(\frac{2}{\delta})}{\epsilon^2}\right),$$

$$d_i \geq \max\left(\frac{I_1\beta_i}{(\beta_s)_i} + \sqrt{\frac{I_1(\beta_i-1)}{(\beta_s)_i}}, cl_i(u_0)_i \frac{\log(I_1)\log(\frac{4}{\delta})}{\epsilon^2}\right).$$

Then

$$\begin{aligned} \|\mathcal{A} - \mathcal{Y}\|_F &= \frac{1}{\sqrt{I_3}}\|\hat{\mathcal{A}} - \hat{\mathcal{Y}}\|_F = \frac{1}{\sqrt{I_3}}\sum_i\|\hat{\mathcal{A}}^{(i)} - \hat{\mathcal{Y}}^{(i)}\|_F \\ &\leq \frac{1}{\sqrt{I_3}}\sum_i(2 + 3\epsilon)c_\epsilon\sqrt{I_1l_i + d_iI_2}\Delta \\ &\leq (2 + 3\epsilon)c_\epsilon\sqrt{I_3 \times \max(I_1l_i + d_iI_2)}\Delta \\ &\leq (2 + 3\epsilon)c_\epsilon\sqrt{I_1lI_3 + dI_2I_3}\Delta \end{aligned}$$

with the probability of at least

$$\begin{aligned} \max\left((1 - \delta)(1 - \delta - 0.2)(1 - 10\log(\bar{n})\bar{n}^{-2\beta_i})\right) \\ = (1 - \delta)(1 - \delta - 0.2)(1 - 10\log(\bar{n})), \end{aligned}$$

which completes the proof.

Theorem 3 requires the number of sampled columns and rows used for the accurate recovery of $\hat{\mathcal{A}}^{(i)}$ to decrease as the over-sampling parameter β_s increases with I_1 and I_2 . In the best case, $\beta_s = \Theta(\frac{I_1I_2}{(I_1+I_2)\log^2(I_1+I_2)})$, Theorem 3 requires only $\Theta(\frac{I_1}{I_2}\log^2(I_1+I_2))$ sampled columns and $\Theta(\frac{I_1}{I_2}\log^2(I_1+I_2))$ sampled rows [32]. In the worst case, $\beta_s = \Theta(1)$, and Theorem 3 requires the number of sampled columns and

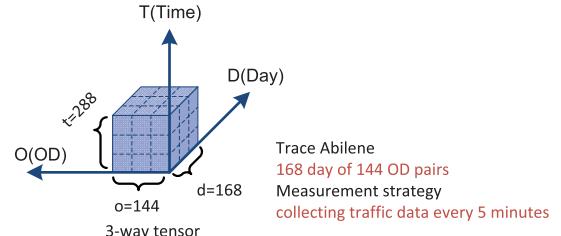


FIGURE 8. Traffic tensor model.

rows to grow linearly with the dimensions of $\hat{\mathcal{A}}^{(i)}$. As a more realistic intermediate scenario, consider the setting in which $\beta_s = \Theta(\sqrt{I_1 + I_2})$ and thus a vanishing fraction of entries are revealed. In this setting, only $O(\sqrt{I_1 + I_2})$ columns and rows are required by Theorem 3.

VII. SIMULATION AND EXPERIMENT

In this section, we use two public real traffic data Abilene [37], GÉANT [38], and one simulated three-way data to evaluate the effectiveness and efficiency of our proposed algorithm.

A. DATA SETS

1) TRAFFIC DATA

Abilene traffic data [37] consists of 12 nodes in cities all over the United States. There are total of 144 OD pairs flows. The time interval of collecting traffic data of Abilene is 5 minutes. GÉANT traffic data [38] is the pan-European research network and composed of 23 routers. The time interval of collecting traffic data of GÉANT is 15 minutes.

Current traffic interpolation approaches usually model the traffic data with a traffic matrix $A \in \mathbb{R}^{o \times \Gamma}$ ($o = N \times N$), where a column of A represents the traffic data of all OD pairs at one time slot. A row of A represents the time evolution of a single OD pair. However, modeling the traffic data in the matrix format cannot sufficiently capture the similarity characteristics of the traffic data. To fully exploit the traffic features, we model the traffic data as a 3-way tensor $\mathcal{A} \in \mathbb{R}^{o \times t \times d}$, where $o = N \times N$. There are d days present with each day having t time intervals. For Abilene data, $t = 288$, $o = 144$ and $d = 168$. For GÉANT data, we selected its subset to get a complete data set, and finally get $t = 96$, $o = 100$ and $d = 107$. Fig.8 shows an illustration of Abilene. In our traffic data model, the random sampling strategy is unfavorable because it changes often (daily), hence the tubal sampling is more suitable.

2) SIMULATED DATA

We use the *create_problem* function of the Tensor Toolbox [39] to create our simulated data. The *create_problem* function allows a user to generate a test problem with a pre-specified solution. It can generate an example data tensor optionally corrupted with standard norm distribution noise. We set the size of data tensor to be $200 \times 200 \times 200$, set the rank to be 10, and set the noise to be 10%. We create our

simulated data by $\text{info} = \text{create_problem}([200 200 200], 10, 0.1)$ and $\mathcal{A} = \text{info}.\text{Data}$. Then \mathcal{A} is a $200 \times 200 \times 200$ tensor with rank = 10 and 10% noise.

B. COMPUTATIONAL ENVIRONMENT AND METHODS

Our method was compared with other three state-of-the-art tensor completion methods

- 1) CP_{als} : CP_{als} addresses the problem of fitting the CP model to incomplete data sets by solving an alternating least-square problem. It was implemented using the Tensor Toolbox [39].
- 2) TK_{als} : TK_{als} addresses the problem of fitting the Tucker model to incomplete data sets by solving an alternating least-squares problem. It was implemented using the Tensor Toolbox.
- 3) T-SVD: T-SVD addresses the problem of fitting the tensor SVD model to incomplete data sets by solving a convex optimization problem.

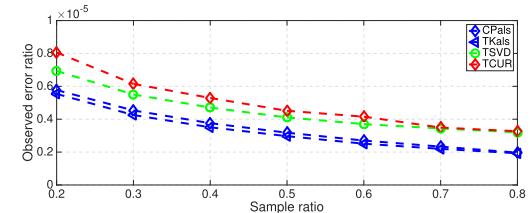
All the simulations were performed with MATLAB R2015b on a MacBook Pro, equipped with Intel Core i5 CPUs (2.90GHz) and 8.00GB RAM. For CP_{als} and TK_{als} , we set the tensor rank to 10, and the max iteration to 500. For T-SVD and T-CUR, we set the max iteration to 50. Following the Theorem 3, we set both the value of sampled columns and sampled rows as $\log^2(I_1 + I_2)$ for T-CUR. The same set of starting points was used by all the methods in the same order. For traffic data, we set parameter δ of our base MC algorithm (SVT) to 1.2 and set parameter τ to 10. For simulated data, we set parameter δ to 1.2 and set parameter τ to default.

We then deployed random tubal sampling and compared the results of recovery at different sampling ratios for all methods. The estimated performance was evaluated by Observed Error Ratio, Recovery Error Ratio and Recovery Computation Time. The formulation is defined as

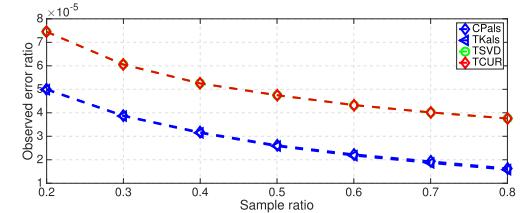
Definition 17 (Observed Error Ratio): A metric for measuring the recovery error of entries in the tensor after the interpolation, which can be calculated as $\text{ObservedError} = \frac{1}{T} \sqrt{\sum_{(i,j,k) \in \Omega} \frac{(x_{ijk} - \hat{x}_{ijk})^2}{x_{ijk}^2}}$, where $1 \leq i \leq o$, $1 \leq j \leq t$ and $1 \leq k \leq d$. x_{ijk} and \hat{x}_{ijk} denote the raw data and the recovered data at (i, j, k) -th element of \mathcal{X} , respectively. T is the total data entries, that is, $T = o \times t \times k$. Note that only entry observed (i.e., $(i, j, k) \in \Omega$) is counted in observed error ratio.

Definition 18 (Recovery Error Ratio): A metric for measuring the recovery error of entries in the tensor after the interpolation, which can be calculated as $\text{RecoveryError} = \frac{1}{T} \sqrt{\sum_{(i,j,k) \in \bar{\Omega}} \frac{(x_{ijk} - \hat{x}_{ijk})^2}{x_{ijk}^2}}$ where $1 \leq i \leq I$, $1 \leq j \leq J$ and $1 \leq k \leq K$. Different from observed error ratio, only the entry not observed is counted in the metric, that is, $(i, j, k) \in \bar{\Omega}$ is calculated.

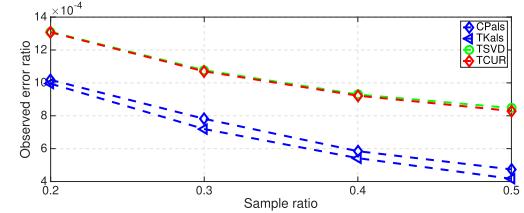
Definition 19 (Recovery Computation Time): A for measuring the number of seconds of one sequence recovery step.



(a)



(b)



(c)

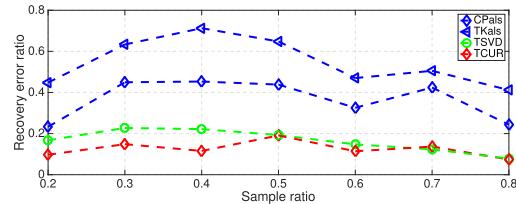
FIGURE 9. Observed Error Ratio. (a) Abilene. (b) GÉANT. (c) Simulated data.

C. RESULTS ANALYSIS

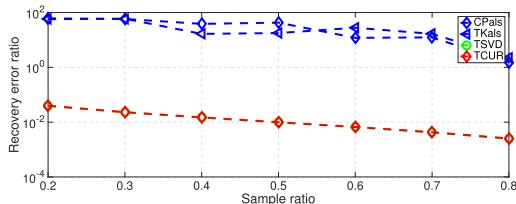
In this subsection, we first present the simulation results on error ratio comparison, and then present the recovery computation time comparison.

In Fig.9, we use the real traffic data and simulated data respectively to compare the observed error ratio under different sample ratios. In all subfigures, we found that the observed error ratios of the four schemes are all under 14×10^{-4} , which is a very small value. These results demonstrate that all schemes can achieve a good observed error ratio performance in both real traffic data and simulated data experiments. Moreover, compared to T-SVD, our T-CUR has almost the same observed error ratio at different sampling rates. These results demonstrate that our T-CUR scheme is instrumental on achieving observed error ratio performance as good as T-SVD.

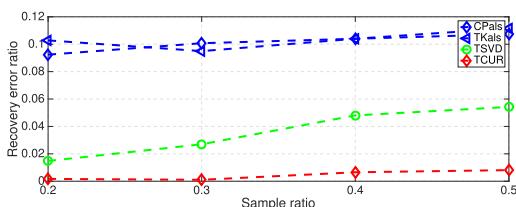
Fig.10 shows the recovery error ratio under different sample ratio. Compared to CP_{als} and TK_{als} , T-SVD and T-CUR in all subfigures have much lower recovery error ratio even though all schemes have the same sampling ratio. This demonstrates that by using t-product, T-SVD and T-CUR becomes effective and are able to capture the global



(a)



(b)



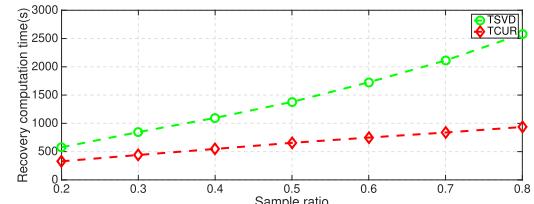
(c)

FIGURE 10. Recovery Error Ratio. (a) Abilene. (b) GÉANT. (c) Simulated data.

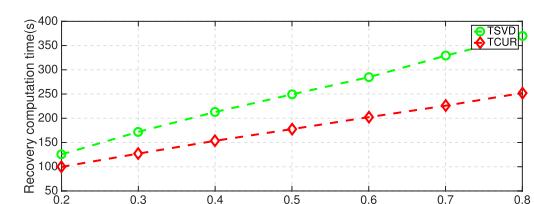
information in the data tensor to recover the missing data with a higher accuracy. As shown in Fig.10(b), the recovery error ratio under CP_{als} and TK_{als} raise up to 100, which is a very large value. We can conclude that CP_{als} and TK_{als} can not recovers the missing data when the whole tubal column is missing. Moreover, compared to T-SVD, we found that the recovery error ratios under T-CUR are not more than those under T-SVD. In particular, as shown in Fig.10(c), the recovery error ratio under T-CUR is much lower than that under T-SVD, which further demonstrates the effectiveness of our T-CUR scheme in supporting high accuracy tensor decomposition.

Although T-CUR and T-SVD can achieve similar recovery accuracy performance on error ratios, in the next paragraph, we will show that the recovery computation time under T-CUR is much faster than that under T-SVD.

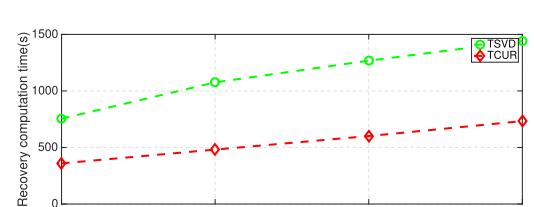
Fig.11 compares the recovery computation time under T-SVD and T-CUR. As expected, the recovery computation time increases with the increase on sampling ratio. This is because more data are involved in the computation when sampling ratio increases. Compared to T-SVD, we found that the recovery computation time under T-CUR is much



(a)



(b)



(c)

FIGURE 11. Recovery computation time. (a) Abilene. (b) GÉANT. (c) Simulated data.

lower than that under T-SVD. As shown in Fig.11(a), when the sampling ratio reaches 80%, the recovery computation time of the T-SVD raises up to 2500s, while the recovery computation time under our T-CUR only raises up to 1000s. T-CUR requires only 40% of the total time needed by T-SVD. These results demonstrate that our T-CUR is a very effective method for reducing the computation time. Moreover, with the increase of sampling ratio, the recovery computation time gap between T-SVD and our T-CUR becomes larger, which means the recovery computation time under T-SVD increases much faster than that under T-CUR.

All these simulation results demonstrate that our T-CUR is a very effective scheme, that is able to achieve very good recovery performance with lower observed error ratio, recovery error ratio and a very short computation time.

VIII. CONCLUSION

In this paper, we formulated the missing data recovery problem of a 3-way tensor as a tensor completion problem. We proposed a novel tensor completion algorithm to quickly recover the missing data from limited samples.

It is a new way to generalize the matrix-CUR into a 3-way tensor. Meaning that a 3-way tensor can now be written as a product of 3-way tensors. We have done extensive simulations on both real data set and simulated data set to verify the efficiency and effectiveness of our proposed algorithm. Compared with other methods in the literature, our algorithm can achieve very good recovery performance, based on the observed error ratio, recovery error ratio and computation time.

REFERENCES

- [1] E. Sejdic, M. A. Rothfuss, M. L. Gimbel, and M. H. Mickle, "Comparative analysis of compressive sensing approaches for recovery of missing samples in implantable wireless Doppler device," *IET Signal Process.*, vol. 8, no. 3, pp. 230–238, 2014.
- [2] S. K. Perepu and A. K. Tangirala, "Reconstruction of missing data using compressed sensing techniques with adaptive dictionary," *J. Process Control*, vol. 47, pp. 175–190, Nov. 2016.
- [3] K. Xie *et al.*, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Proc. IEEE Int. Conf. Comput. Commun.*, Apr. 2015, pp. 2443–2451.
- [4] S. Yang *et al.*, "Online recovery of missing values in vital signs data streams using low-rank matrix completion," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2013, pp. 281–287.
- [5] K. Xie, L. Wang, X. Wang, J. Wen, and G. Xie, "Learning from the past: Intelligent on-line weather monitoring based on matrix completion," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2014, pp. 176–185.
- [6] M. Mani, M. Jacob, D. Kelley, and V. Magnotta. (Feb. 2016). "Multi-shot multi-channel diffusion data recovery using structured low-rank matrix completion." [Online]. Available: <https://arxiv.org/abs/1602.07274>
- [7] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. ACM SIGMETRICS*, 2003, pp. 61–72.
- [8] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, pp. 947–960, Oct. 2005.
- [9] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 219–230, 2004.
- [10] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.
- [11] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and Internet traffic matrices (extended version)," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.
- [12] G. Gürsun and M. Crovella, "On traffic matrix completion in the Internet," in *Proc. ACM IMC*, 2012, pp. 399–412.
- [13] J. A. Bengua, H. N. Phiem, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.
- [14] N. Li and B. Li, "Tensor completion for on-board compression of hyperspectral images," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 517–520.
- [15] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. ACM Conf. Recommender Syst. (RecSys)*, Barcelona, Spain, Sep. 2010, pp. 79–86.
- [16] W. W. Sun, J. Lu, H. Liu, and G. Cheng, "Provable sparse tensor decomposition," *J. Roy. Stat. Soc.*, vol. 79, no. 3, pp. 899–916, 2016.
- [17] M. Tang, G. Ding, Q. Wu, Z. Xue, and T. A. Tsiftsis, "A joint tensor completion and prediction scheme for multi-dimensional spectrum map construction," *IEEE Access*, vol. 4, no. 99, pp. 8044–8052, 2016.
- [18] Y. Zhang, C. D. Silva, R. Kumar, and F. J. Herrmann, "Massive 3D seismic data compression and inversion with hierarchical tucker," in *Proc. SEG*, 2017, pp. 1347–1352.
- [19] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
- [20] B. Ran, H. Tan, Y. Wu, and P. J. Jin, "Tensor based missing traffic data completion with spatial-temporal correlation," *Phys. A, Stat. Mech. Appl.*, vol. 446, pp. 54–63, Mar. 2016.
- [21] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011.
- [22] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 15–27, Mar. 2013.
- [23] J. Håstad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.
- [24] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover, "Facial recognition using tensor-tensor decompositions," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 437–463, 2013.
- [25] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.
- [26] L. de Lathauwer, B. de Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [27] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [28] M. E. Kilmer, C. D. Martin, and L. Perrone, "A third-order generalization of the matrix svd as a product of third-order tensors," Dept. Comput. Sci., Tufts Univ., Tech. Rep. TR-2008-4, 2008.
- [29] J. H. de M. Goulart, M. Boizard, R. Boyer, G. Favier, and P. Comon, "Tensor CP decomposition with structured factor matrices: Algorithms and performance," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 757–769, Jun. 2016.
- [30] P. Jain and S. Oh, "Provable tensor factorization with missing data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1431–1439.
- [31] C. Lubich, T. Rohwedder, R. Schneider, and B. Vandereycken, "Dynamical approximation by hierarchical tucker and tensor-train tensors," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 2, pp. 470–494, 2013.
- [32] L. W. Mackey, M. I. Jordan, and A. Talwalkar, "Divide-and-conquer matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1134–1142.
- [33] Y. Pi, H. Peng, S. Zhou, and Z. Zhang, "A scalable approach to column-based low-rank matrix approximation," in *Proc. IJCAI*, 2013, pp. 1600–1606.
- [34] H. Ye, Y. Li, and Z. Zhang. (Nov. 2015). "A simple approach to optimal CUR decomposition." [Online]. Available: <https://arxiv.org/abs/1511.01598>
- [35] M. Xu, R. Jin, and Z.-H. Zhou. (Nov. 2015). "CUR algorithm for partially observed matrices." [Online]. Available: <https://arxiv.org/abs/1411.0860>
- [36] J.-F. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [37] 2004 *Abilene Data*. Accessed: Apr. 10, 2017 [Online]. Available: http://www.maths.adelaide.edu.au/matthew.roughan/project/traffic_matrix/
- [38] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 83–86, 2006.
- [39] B. W. Bader *et al.* (2012). *MATLAB Tensor Toolbox Version 2.5*. [Online]. Available: <http://www.sandia.gov/~tgkolda/TensorToolbox/>



LELE WANG is currently working toward the Ph.D. degree at Hunan University, Changsha, China.



KUN XIE received the Ph.D. degree in computer application from Hunan University, Changsha, China, in 2007. She was a Postdoctoral Fellow at the Department of Computing, Hong Kong Polytechnic University, from 2007 to 2010. She was a Visiting Researcher at the Department of Electrical and Computer Engineering, State University of New York at Stony Brook, from 2012 to 2013. She is currently a Professor at Hunan University. She has authored or coauthored over 60 papers in major journals and conference proceedings including the journals (the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, and the IEEE TRANSACTIONS ON COMPUTERS) and conferences (INFOCOM, ICDCS, SECON, and IWQoS). Her research interests include wireless network and mobile computing, network management and control, cloud computing and mobile cloud, and big data.



HUIBIN ZHOU received the Ph.D. degree in computer application from Hunan University in 2016. His current research interests include dependable systems/networks, network measurement, network management, and software-defined networking.

• • •



THABO SEMONG is currently working toward the Ph.D. degree at Hunan University, Changsha, China.