Accurate Recovery of Internet Traffic Data Under Variable Rate Measurements

Kun Xie, Can Peng, Xin Wang, *Member, IEEE*, Gaogang Xie, Jigang Wen, Jiannong Cao, *Fellow, IEEE*, Dafang Zhang, and Zheng Qin

***Abstract*— The inference of the network traffic matrix from partial measurement data becomes increasingly critical for various network engineering tasks, such as capacity planning, load balancing, path setup, network provisioning, anomaly detection, and failure recovery. The recent study shows it is promising to more accurately interpolate the missing data with a 3-D tensor as compared with the interpolation methods based on a 2-D matrix. Despite the potential, it is difficult to form a tensor with measurements taken at varying rate in a practical network. To address the issues, we propose a Reshape-Align scheme to form the regular tensor with data from variable rate measurements, and introduce user-domain and temporal-domain factor matrices which take full advantage of features from both domains to translate the matrix completion problem to the tensor completion problem based on CANDECOMP/PARAFAC decomposition for more accurate missing data recovery. Our performance results demonstrate that our Reshape-Align scheme can achieve significantly better performance in terms of several metrics: error ratio, mean absolute error, and root mean square error.**

***Index Terms*— Internet traffic data recovery, matrix completion, tensor completion.**

I. Introduction

A TRAFFIC matrix (TM) is often applied to track the volume of traffic between origin-destination (OD) pairs in a network. Estimating the end-to-end TM in a network is an essential part of many network design and traffic engineering tasks, including capacity planning, load balancing, path setup, network provisioning, anomaly detection, and failure recovery.

Due to the lack of measurement infrastructure, direct and precise end-to-end flow traffic measurement is extremely difficult in the traditional IP network [1]. Thus previous work on TM estimation focus on inferring the TM indirectly from link loads [2], [3], and the methods taken are often sensitive to the statistical assumptions made for models and the TMs estimated are subject to large errors [4].

As an alternative, TM is directly built through the collection of the end-to-end flow-level traffic information using flow monitoring tools such as Cisco NetFlow, and the recent OpenFlow [5]. Unlike commodity switches in traditional IP networks, flow-level operations are streamlined into OpenFlow switches, which provides the possibility of querying and obtaining the end-to-end flow traffic statistics. Despite the progress in flow-level measurements, the collection of the traffic information network wide to form TM at fine time scale still faces many challenges:

- Due to the high network monitoring and communication cost, it is impractical to collect full traffic volume information from a very large number of points. Sample-based traffic monitoring is often applied where measurements are only taken between some random node pairs or at some of the periods for a given node pair.
- Measurement data may get lost due to severe communication and system conditions, including network congestion, node misbehavior, monitor failure, transmission of measurement information through an unreliable transport protocol.

As many traffic engineering tasks (such as anomaly detection, traffic prediction) require the complete traffic volume information (i.e., the complete traffic matrix) or are highly sensitive to the missing data, the accurate reconstruction of missing values from partial traffic measurements becomes a key problem, and we refer this problem as the traffic data recovery problem.

Various studies have been made to handle and recover the missing traffic data. Designed based on purely spatial [6]–[8] or purely temporal [9], [10] information, the data recovery performance of most known approaches is low. Recently matrix-completion-based algorithms are proposed to recover the missing traffic data by exploiting both spatial and temporal information [11]–[15]. Although the performance is good when the data missing ratio is low, the performance suffers when the missing ratio is large.

Manuscript received April 24, 2017; revised November 25, 2017 and February 5, 2018; accepted March 6, 2018; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Joo. Date of publication April 18, 2018; date of current version June 14, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61572184, Grant 61725206, Grant 61472130, Grant 61472131, and Grant 61772191, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2017JJ1010, in part by the Science and Technology Key Projects of Hunan Province under Grant 2015TP1004 and Grant 2016JC2012, in part by the U.S. ONR under Grant N00014-17-1-2730, in part by the NSF under Grant ECCS 1408247, Grant CNS 1526843, and Grant ECCS 1731238, and in part by the Open Project Funding of the CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, under Grant CASNDST201704. *(Corresponding author: Kun Xie.)*

K. Xie is with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410006, China, and also with the CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: xiekun@hnu.edu.cn).

C. Peng, D. Zhang, and Z. Qin are with the College of Computer Science and Electronics Engineering, Hunan University, Changsha 410006, China (e-mail: pengcancaroline@gmail.com; dfzhang@hnu.edu.cn; zqin@hnu.edu.cn).

X. Wang is with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY 11794 USA (e-mail: x.wang@stonybrook.edu).

G. Xie and J. Wen are with the Network Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xie@ict.ac.cn; wenjigang@ict.ac.cn).

J. Cao is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong (e-mail: csjcao@comp.polyu.edu.hk).

Digital Object Identifier 10.1109/TNET.2018.2819504

Based on the analyses of real traffic trace, our recent work in [16] reveals that the traffic data have the features of temporal stability, spatial correlation, and periodicity. Specially, the periodicity features indicate that users usually have similar Internet visiting behaviors at the same time of a day, so the measurements for an OD pair taken at the same time slots of two consecutive days are similar. For more accurate missing data interpolation, we can take advantage of these features to concurrently consider the traffic of different days and model the traffic data as a 3-way tensor. Thus, we can infer the complete traffic matrices of multiple days through tensor completion [16].

Tensors are the higher-order generalization of vectors and matrices. Tensor-based multilinear data analysis has shown that tensor models can take full advantage of the multilinear structures to provide better data understanding and information precision. Tensor-based analytical tools have seen applications for web graphs [17], knowledge bases [18], chemometrics [19], signal processing [20], traffic data management in transportation [21], [22], and computer vision [23], etc.

Compared with matrix-based data recovery, the tensor-based approach can better handle the missing traffic data and will be used in this paper. Although promising, the traffic tensor model in [16], [24], and [25] is built with a strong assumption that the network monitoring system adopts a static measurement strategy by taking traffic samples at a fixed rate. However, in a practical network monitoring system, the rate of measurements is often adapted according to the traffic conditions (i.e., varying in different periods of a day) and some traffic engineering requirements (i.e., to more timely detect anomaly). The variable rate measurements make it hard to form a regular traffic tensor for further processing. Some challenges due to the variation of the measurement rate are:

- ***Difficult to align the matrices of different days.*** The traffic matrices of different days would have different number of columns, which makes it hard to integrate the traffic matrices of different days to form a standard tensor and recover the missing data.
- ***Difference in the length of the time slot.*** The sample data in a column of the traffic matrix may correspond to a time slot with a different length, which further brings the difficulty of recovering the missing items through the temporal and spatial correlation among traffic data.

Despite the challenges, the traffic matrix has some special features: 1) The traffic matrices of different days record the data of the same OD pairs in the network, and 2) The user traffic data follow a daily schedule. Therefore, there should exist some common *user-domain* and *time-domain* features that can be exploited for more accurate interpolation.

In this paper, we propose a novel traffic data recovery scheme in the presence of variation of traffic measurement rate. Our scheme will first construct a regular tensor with the reshaping and alignment of traffic matrices with inconsistent number of columns and different length of time slots, and then enable more accurate traffic data recovery taking advantage of the data correlation in a three dimensional tensor. The contributions of this paper can be summarized as follows:

- We propose a matrix division algorithm for time alignment, which exploits our novel time rule to efficiently divide the traffic matrices into sub-matrices with each corresponding to one time segment with the same sampling rate.
- We reshape and align traffic matrices from measurements with variable rates to form a regular tensor, taking advantage of multi-dimensional data correlation for more accurate traffic data recovery. To address the challenge of integrating matrices of different dimensions into a tensor, we introduce user-domain and temporal-domain factor matrices to translate the problem of matrix completion for different days to the problem of tensor completion based on CANDECOMP/PARAFAC (CP) decomposition [26], [27].
- We compare the proposed Reshape-Align scheme with the state of art matrix-completion and tensor completion algorithms, and our results demonstrate that our scheme can achieve significantly better performance in terms of several metrics: error ratio, mean absolute error (MAE), and root mean square error (RMSE).

To the best of our knowledge, our Reshape-Align scheme is the first one that considers the traffic recovery problem under variable rate measurements in a practical network system, and provides a novel reshaping and alignment technique that allows the integration of inconsistent traffic matrices to form a standard tensor for more accurate missing data recovery.

The rest of the paper is organized as follows. We introduce the related work in Section II. The preliminaries of tensor are presented in Section III. We present the problem and our overview solution in Section IV. The proposed algorithms on matrix division for time alignment, and matrix reshaping and alignment for tensor completion are presented in Section V and Section VI, respectively. Finally, we evaluate the performance of the proposed algorithm through extensive simulations in Section VIII, and conclude the work in Section IX.

II. Related Work

A set of studies have been made to handle the missing traffic data. Designed based on purely spatial [6]–[8] or purely temporal [9], [10] information, most of the known approaches have a low data recovery performance.

To capture more spatial-temporal features in the traffic data, SRMF [11] proposes the first spatio-temporal model of traffic matrices (TMs). To recover the missing data, SRMF is designed based on low-rank approximation combined with the spatio-temporal operation and local interpolation. Following SRMF, several other matrix recovery algorithms [12]–[15], [28]–[31] are proposed to recover the missing data from partial traffic or network latency measurements. Compared with the vector-based recovery approaches [6]–[10], as a matrix could capture more information and correlation among traffic data, matrix-based approaches achieve much better recovery performance.

However, a two-dimension matrix is still limited in capturing a large variety of correlation features hidden in the traffic data. For example, although the traffic matrix defined in [11]

can catch the spatial correlation among flows and the small-scale temporal feature, it can not incorporate other temporal features such as the feature of the traffic periodicity cross day. Therefore, a matrix is still not enough to capture the comprehensive correlations among the traffic data, and the data recovery performance under the matrix-based approaches can be improved.

To further utilize the traffic periodicity feature for accurate traffic data recovery, the recent studies [16], [24] combine the traffic matrices of different days to form a tensor to recover the missing data. Several tensor completion algorithms [32]–[35] are proposed for recovering the missing data by capturing the global structure of the data via a high-order decomposition (such as CANDECOMP/PARAFAC (CP) decomposition [26], [27] and Tucker decomposition [36]). Tensor has proven to be good data structure for dealing with the multi-dimensional data in a variety of fields [17]–[23]. Very few recent studies [16], [24], [25], [37], [38] begin to deal with Internet data such as interfering the missing data and detecting anomaly through tensor decomposition, among which, [16], [24], [25], [37] are our previous studies. Although promising, the traffic tensor model in these studies is built with a strong assumption that the network monitoring system adopts a static measurement strategy with a fixed rate to take data samples. The proposed methods may fail to work in a practical network monitoring scenario where the rate of measurements varies over time.

To address this practical challenge, we propose a novel Reshape-Align scheme with several novel techniques, including matrix division for time alignment, mechanism to reshape and align matrices, and the technique to solve the matrix completion problem through tensor CP decomposition. The simulation results demonstrate that Reshape-Align scheme can achieve significantly better performance in terms of several metrics: error ratio, mean absolute error (MAE), and root mean square error (RMSE).

III. Preliminaries

The notation used in this paper is described as follows. Scalars are denoted by lowercase letters $(a, b, \cdots)$, vectors are written in boldface lowercase $(\mathbf{a}, \mathbf{b}, \cdots)$, and matrices are represented with boldface capitals $(\mathbf{A}, \mathbf{B}, \cdots)$. Higher-order tensors are written as calligraphic letters $(\mathcal{X}, \mathcal{Y}, \cdots)$. The elements of a tensor are denoted by the symbolic name of the tensor with indexes in subscript. For example, the ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i, j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, and element (i, j, k) of a third-order tensor $\mathcal{X}$ is denoted by x_{ijk}.

Definition 1: A tensor *is a multidimensional array, and is a higher-order generalization of a vector (first-order tensor) and a matrix (second-order tensor). An N-way or Nth-order tensor (denoted as $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$) is an element of the tensor product of N vector spaces, where N is the order of $\mathcal{A}$, also called way or mode.*

The element of $\mathcal{A}$ is denoted by $a_{i_1, i_2, \cdots, i_N}$, $i_n \in \{1, 2, \cdots, I_n\}$ with $1 \leq n \leq N$.

Definition 2: Slices are two-dimensional sub-arrays, defined by fixing all indexes but two.

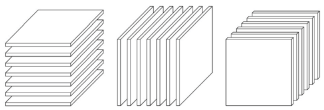


Fig. 1. Tensor slices.

In Fig. 1, a 3-way tensor $\mathcal{X}$ has horizontal, lateral and frontal slices, which are denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$ and $\mathbf{X}_{::k}$, respectively. In this paper, we denote the frontal slice $\mathbf{X}_{::k}$ as $\mathbf{X}_k$.

Definition 3: The outer product of two vectors $\mathbf{a} \circ \mathbf{b}$ is the matrix defined by: $(\mathbf{a} \circ \mathbf{b})_{ij} = a_i b_j$.

Definition 4: The outer product $\mathcal{A} \circ \mathcal{B}$ of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N_1}}$ and a tensor $\mathcal{B} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_{N_2}}$ is the tensor of the order $N_1 + N_2$ defined by

$$(\mathcal{A} \circ \mathcal{B})_{i_1, i_2, \cdots, i_{N_1}, j_1, j_2, \cdots, j_{N_2}} = a_{i_1, i_2, \cdots, i_{N_1}} b_{j_1, j_2, \cdots, j_{N_2}} \quad (1)$$

for all values of the indexes.

Since vectors are first-order tensors, the outer product of three vectors $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ is a tensor given by:

$$(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})_{ijk} = a_i b_j c_k \quad (2)$$

for all values of the indexes.

Definition 5: A 3-way tensor $\mathcal{X}$ is a rank one tensor if it can be written as the outer product of three vectors, i.e. $\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$.

Definition 6: The rank of a 3-way tensor is the minimal number of rank one tensors, that generate the tensor as their sum, i.e. the smallest R, such that $\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$.

Definition 7: The idea of CANDECOMP/PARAFAC (CP) decomposition is to express a tensor as the sum of a finite number of rank one tensors. A 3-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ can be expressed as

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (3)$$

with an entry calculated as

$$x_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \quad (4)$$

where $R > 0$, a_{ir}, b_{jr}, c_{kr} are the i-th, j-th, and k-th entry of vectors $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$, respectively.

By collecting the vectors in the rank one components, we have tensor $\mathcal{X}$'s factor matrices $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}$, $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}$, and $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_R] \in \mathbb{R}^{K \times R}$. Using the factor matrices, we can rewrite the CP decomposition as follows.

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \quad (5)$$

Fig. 2 illustrates the CP decomposition. In this paper, we design traffic data recovery algorithm based on the CP decomposition.

IV. Problem Description and Solution Overview

As the matrix size of each day is different due to variable rate measurements, we cannot use tensor to recover the missing data directly. In this section, we first formulate the traffic data recovery problem as a matrix factorization problem, and then present the benefit and methodology of transforming

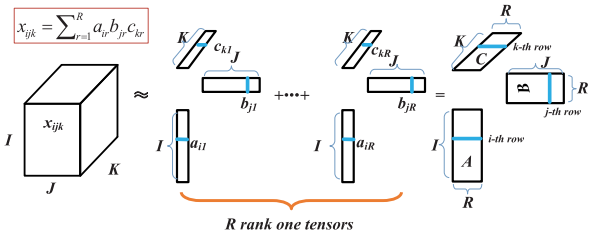


Fig. 2. CP decomposition of three-way tensor as sum of R outer products (rank one tensors). CP decomposition can be written as a triplet of factor matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, i.e, the r-th column of which contains $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$, respectively. The entry x_{ijk} can be calculated as the sum of the product of the entries of the i-th row of the matrix $\mathbf{A}$, the j-th row of the matrix $\mathbf{B}$, and the k-th row of the matrix $\mathbf{C}$.

this problem further to the tensor factorization problem along with the difficulty of this transformation in a practical network monitoring system.

A. Empirical Study With Real Traffic Data

In order to infer the missing monitoring data with matrix completion, the rank of the matrix has to be low. We first validate that Internet traffic data have the low rank feature.

For a network consisting of N nodes, there are $n = N \times N$ OD pairs. We define a monitoring data matrix, $\mathbf{X}_k \in \mathbb{R}^{n \times m_k}$, to hold the traffic data measured in the kth day for $k = 1, 2, \cdots, K$. m_k is the total number of time slots captured in the kth day. In the matrix, a row corresponds to an OD pair, a column corresponds to a time slot, and the (ij)-th entry $x_{k:ij}$ represents the monitoring data of the OD pair i at the time slot j.

According to the singular value decomposition (SVD), the rank of a matrix $\mathbf{X}$ (denoted by r) is equal to the number of its non-zero singular values. In this paper, we call this as the "precise rank". Although this rank definition is of high theoretical interest, it is not realistic to use this definition for the practical data. The calculation of the precise rank of the matrix is an ill-posed problem in a practical environment because arbitrary small perturbations of matrix elements may change the rank [39].

According to PCA (Principal components analysis), if a matrix has low-rank, its top k singular values constitute the total variance, that is, $\sum_{i=1}^{k} \sigma_i^2 \approx \sum_{i=1}^{r} \sigma_i^2$, where σ_i is the i-th singular value of the matrix. Consequently, we define the ratio $g(k) = \sum_{i=1}^{k} \sigma_i^2 / \sum_{i=1}^{r} \sigma_i^2$ to indicate what fraction of the total variance (Frobenius norm) in $\mathbf{X}$ is represented by the rank-k truncated SVD of the matrix $\mathbf{X}$. That is, $\mathbf{X} \approx \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$.

We analyze two public traffic traces, Abilene [40] and GÈANT [41]. For each trace, we randomly select data from three days. Fig. 3 plots the fraction of the total variance captured by the top k singular values of the data matrices, with one small figure corresponding to one day. We find that the top 20 singular values capture nearly 100% variance of the monitoring data matrices, which confirms that they are low-rank and provides a prerequisite for using the matrix completion.

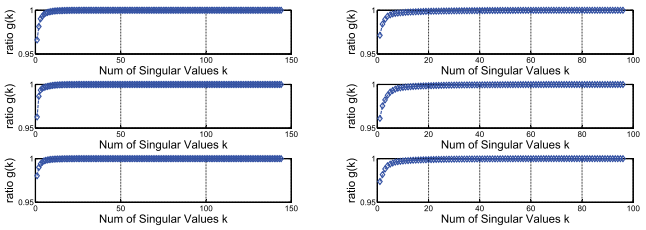


Fig. 3. Fraction captured by top k singular values. (a) Abilene. (b) GÈANT.

According to the results, we set the rank r to preserve 99% of the data variability of the traffic matrix. As matrices of different days record the traffic data of the same network with the same users, different matrices have similar features in user domain and temporal domain, thus similar ranks.

In Section V, we divide data matrices across different days into multiple groups of sub-matrices with the rank r_g denoting the rank of the g-th group. Let $r_1^g, r_2^g, \cdots, r_K^g$ denote the ranks of the sub-matrices in the group g of recent K days. As each sub-matrix records the traffic data of same users in the same time duration of different days, it should have the same features in the user domain and temporal domain, and thus the rank of the sub-matrices in the same group should be same or at least similar if there exist some traffic measurement disturbances. That is, $r_1^g \approx r_2^g \approx \cdots \approx r_K^g$. To reduce the parameter training cost, among the K sub-matrices, we can select l sub-matrices and set r_g to be the maximum rank of the selected sub-matrices. In our simulation, we set $l = 3$.

B. Traffic Recovery Problem Based on Matrix Factorization

To reduce the network monitoring overhead, only a subset of measurements are taken. We apply the matrix factorization to infer the missing entries of the K matrices corresponding to recent K days. A monitoring matrix $\mathbf{X}_k$ with rank r can be factored into a production of an $n \times r$ factor matrix $\mathbf{U}_k$ for the *user domain*, an $r \times r$ diagonal matrix $\mathbf{\Sigma}_k$, and an $m_k \times r$ factor matrix $\mathbf{V}_k$ for the *time domain*. To infer the missing entries of K matrices, we can minimize the loss function as follows:

$$\min_{\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k} f(\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k)$$
$$\text{s.t. } f(\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k) = \frac{1}{2} \sum_{k=1}^{K} \left\| \left(\mathbf{X}_k - \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \right)_{\Omega_k} \right\|_F^2 \tag{6}$$

where $f(\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k)$ is the loss function defined based on the Frobenius norm $\|\|_F$ and Ω_k is the index set of the observed samples of the matrix $\mathbf{X}_k$.

After obtaining the factor matrix $\mathbf{U}_k$, the diagonal matrix $\mathbf{\Sigma}_k$, and the factor matrix $\mathbf{V}_k$, the monitoring matrix can be recovered as follows:

$$\hat{\mathbf{X}}_k = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \tag{7}$$

where $\hat{\mathbf{X}}_k$ denotes the recovered traffic matrix.

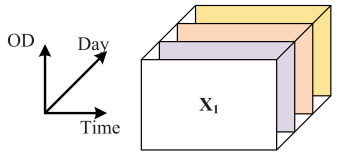


Fig. 4. Tensor based traffic model.

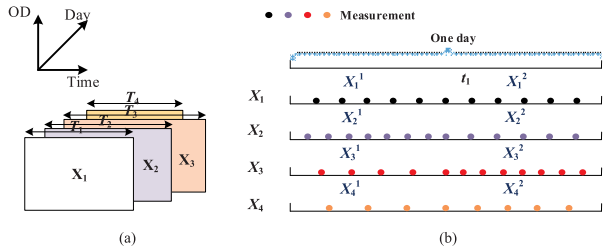


Fig. 5. Traffic matrices with inconsistent number of columns.

C. From Matrix Factorization to Tensor Factorization

As traffic data are observed to possess the features of temporal stability, spatial correlation, and periodicity features [16], [24], rather than only recovering the data through the two dimensional matrix, it is promising to more accurately interpolate the missing data with a three-dimensional tensor (Fig. 4) taking advantage of the periodicity feature of traffic across days. Despite the potential, in a practical network monitoring system, the measurement strategy may vary according to the traffic conditions. There exist some challenges to combine multiple matrices to form a regular tensor:

- ***Inconsistent number of columns across the matrices***. As a column represents a sample in a time slot, the variation of measurement rate in different days would make their traffic matrices to have different number of columns (Fig. 5(a)). This introduces the challenge to forming the standard tensor with these matrices.
- ***Inconsistent length of time slot within the matrix***. Different measurement rate makes columns in a matrix to correspond to different time-slot lengths (as shown in Fig. 5(b)), which further brings the difficulty of recovering the missing items through the temporal and spatial correlation among traffic data.

D. Characteristics in Multiple Data Matrices

Although the variation of the measurement rate brings the challenge of integrating the measurement matrices of different days to form a regular tensor, these matrices have some characteristics that can be exploited for more accurate data recovery.

- Traffic matrices of different days record the measurement data of the same OD pairs in the network, and the row indexes of these matrices are the same. Thus these matrices should have some common OD-domain (i.e., user-domain) features, so in (9), we use the same factor matrix $\mathbf{U}^g$ for different traffic matrices.

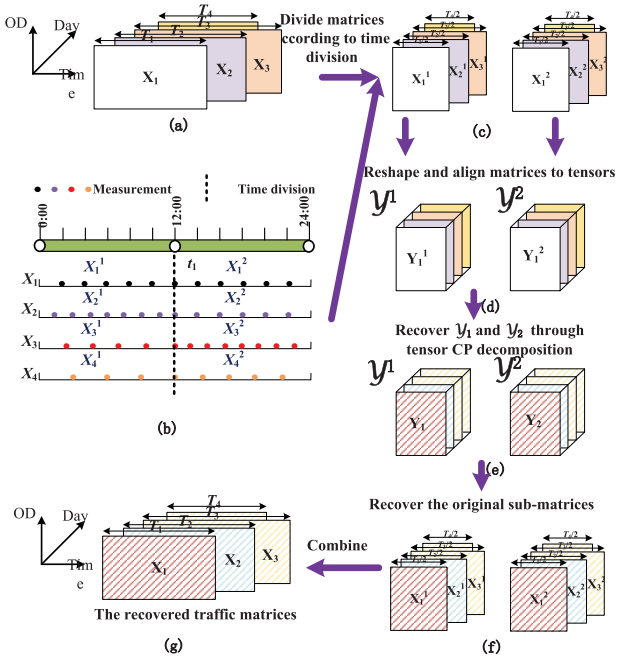


Fig. 6. Overview solution of Reshape-Align scheme.

- Although the number of columns and the time-slot lengths may be different for matrices of different days, the user traffic in these matrices vary following a daily schedule in the temporal domain,

  as users usually have similar Internet access behaviors.

To exploit tensor-based algorithms for more accurate missing data inference, we would like to investigate and take advantage of these common features to reshape and align traffic matrices with inconsistent number of columns and time-slot lengths to form regular tensors. The issues we need to address are: 1) How could we exploit the common time-domain features hidden in the traffic data within a day to align matrices across days? and 2) How to exploit the user domain and temporal domain features to reshape the matrices across days to form the tensor?

E. Solution Overview

To fully exploit the common features hidden in monitoring matrices for more accurate missing data interpolation, we propose a matrix reshaping and alignment scheme in the presence of varying network measurement rate.

Fig. 6(a) shows example traffic matrices to recover. The time slots in a matrix may have different lengths. To well exploit the common time-domain features hidden in the traffic data within a day, the matrices should be divided and aligned in the physical time domain as explained in Section V. Accordingly, we propose a matrix division algorithm with the example shown in Fig. 6 (b), where the matrices are divided in temporal domain to satisfy the time alignment requirement. The sub-matrices formed after the division (in Fig. 6 (c)) will be further utilized to form tensors.

To exploit correlations across days for more accurate data recovery, we further translate the factor matrices of each sub-matrix to common ones taking advantage of the user domain

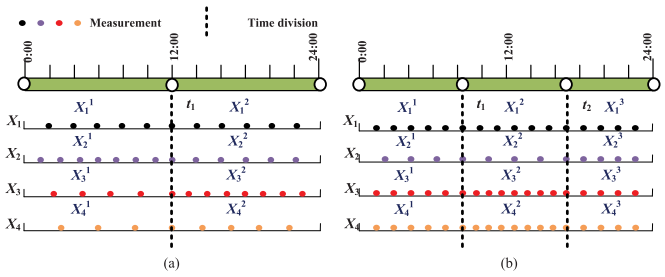


Fig. 7. Time alignment problem.

and temporal domain features hidden in the sub-matrices, and then integrate the reshaped and aligned sub-matrices to form the tensor in Fig. 6 (d). We apply the tensor completion algorithm to interpolate the missing data in Fig. 6 (e), and then take the reverse procedure of reshaping to obtain the recovered sub-matrices (in Fig. 6 (f)), which will be combined to form the final recovered large matrices (in Fig. 6 (g)).

V. Matrix Division for Time Alignment

We first present our matrix division algorithm, then reformulate the recovery problem for the sub-matrices by taking consideration of the common features of matrices in both the user (OD) domain and the time domain.

A. Matrix Division

Although the difference in the traffic measurement rate may result in different time-slot lengths, we can still observe that the user traffic patterns often change daily following the user daily Internet access behaviors. To well exploit the time-domain features hidden in the traffic data, we divide daily measurements into multiple time segments each having a different measurement rate.

Fig. 7 utilizes two examples to illustrate the time alignment problem, where $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, $\mathbf{X}_4$ denote the measurement traffic matrices of four days. In Fig. 7(a), a fixed measurement rate is adopted in $\mathbf{X}_1$ for the whole day. For matrices $\mathbf{X}_2$, $\mathbf{X}_3$, $\mathbf{X}_4$, two different measurement rates are assumed for the first half day and the next half day, respectively. To align the data in the time domain, we divide the whole day time into two time segments, each corresponding to one half day and adopting the same measurement strategy. Accordingly, the original traffic matrices are divided into two parts with $\mathbf{X}_1 = \left[\mathbf{X}_1^1, \mathbf{X}_1^2\right]$, $\mathbf{X}_2 = \left[\mathbf{X}_2^1, \mathbf{X}_2^2\right]$, $\mathbf{X}_3 = \left[\mathbf{X}_3^1, \mathbf{X}_3^2\right]$, $\mathbf{X}_4 = \left[\mathbf{X}_4^1, \mathbf{X}_4^2\right]$. Similarly, in Fig. 7(b), the time in the day is divided into three time segments and thus original traffic matrices are divided into three parts with $\mathbf{X}_1 = \left[\mathbf{X}_1^1, \mathbf{X}_1^2, \mathbf{X}_1^3\right]$, $\mathbf{X}_2 = \left[\mathbf{X}_2^1, \mathbf{X}_2^2, \mathbf{X}_2^3\right]$, $\mathbf{X}_3 = \left[\mathbf{X}_3^1, \mathbf{X}_3^2, \mathbf{X}_3^3\right]$, $\mathbf{X}_4 = \left[\mathbf{X}_4^1, \mathbf{X}_4^2, \mathbf{X}_4^3\right]$.

B. Time Division and Alignment

As the network monitoring center has the knowledge of the measurement strategy, it can easily perform such time division and alignment. To facilitate the alignment process, the center will record the changes of measurement rate in different days. When the measurement rate is largely changed in a day, the center inserts a time spot with the date of the record and the measurement rates before and after the time spot. Obviously, the time duration between two time spots corresponds to one segment, and thus the time division can be easily obtained based on the inserted time spots. Fig. 8 shows the time spot of Fig. 7(b).

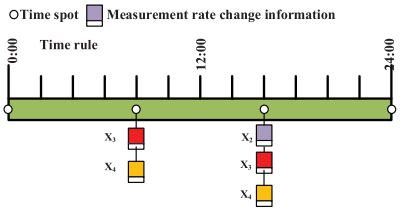


Fig. 8. A time rule to record measurement rate change. This example shows the time rule state of Fig. 7(b). Besides the start (0:00) and the end (24:00), there are two time spots in the time rule. One links the measurement rate information of the 3rd day (i.e., $\mathbf{X}_3$) and the fourth day (i.e., $\mathbf{X}_4$) as these two days change their measurement rates at this time spot as shown in Fig. 7(b). The other links the measurement rate information of the 2nd day (i.e., $\mathbf{X}_2$), the 3rd day (i.e., $\mathbf{X}_3$), and the fourth day (i.e., $\mathbf{X}_4$). As the first day (i.e., $\mathbf{X}_1$) adopts the static measurement rate in the whole day, $\mathbf{X}_1$ does not correspond to any time spot.

Although continuous network monitoring will provide a sequence of traffic matrices with each for one day, for more accurate recovery and analysis of the traffic data, network engineering tasks usually only consider the traffic data in recent days. In our design, with the recent days selected, a time spot is set to be active only if it is linked to one of the selected days. The duration of a day is divided according to the active time spots only.

As shown in Fig. 7, obviously, the divided sub-matrices $\mathbf{X}_1^1$, $\mathbf{X}_2^1$, $\mathbf{X}_3^1$, and $\mathbf{X}_4^1$ record the traffic data of the same time duration in different days, and can be combined for more accurate traffic data recovery. Our reshaping and alignment scheme exploits these temporal domain features hidden in the matrices to more accurately recover the missing data.

C. Problem Reformulation With Common User and Temporal Domain Features

After the time alignment, the original matrices are divided into multiple sub-matrices. We denote sub-matrices that record the traffic data of the same time segment of different days as one sub-matrix group. If the duration of a day k is partitioned into G time segments, we have $\mathbf{X}_k = [\mathbf{X}_k^1, \mathbf{X}_k^2, \cdots \mathbf{X}_k^G]$ for $k = 1, 2, \cdots, K$. According to the time division and alignment requirement, all matrices are divided at the same time spot to cover the same time segment. Therefore, after the matrix division, there are G groups of sub-matrices with each group having K sub-matrices, that is $\{\mathbf{X}_1^g, \mathbf{X}_2^g, \cdots, \mathbf{X}_K^g\}$ for $g = 1, 2, \cdots, G$.

According to the partition, the problem in (6) can be transformed to the problem of minimizing the total loss from the recovery of all sub-matrices:

$$\begin{aligned} &\min_{\mathbf{U}_k^g, \mathbf{\Sigma}_k^g, \mathbf{V}_k^g} f\left(\mathbf{U}_k^g, \mathbf{\Sigma}_k^g, \mathbf{V}_k^g\right) \\ &\text{s.t. } f\left(\mathbf{U}_k^g, \mathbf{\Sigma}_k^g, \mathbf{V}_k^g\right) \\ &\qquad = \frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{X}_k^g - \mathbf{U}_k^g\mathbf{\Sigma}_k^g(\mathbf{V}_k^g)^T\right)_{\Omega_k^g}\right\|_F^2\right) \end{aligned} \tag{8}$$

where $\mathbf{X}_k^g \in \mathbb{R}^{n\times m_{k:g}}$, $\mathbf{U}_k^g \in \mathbb{R}^{n\times r_g}$, $\mathbf{V}_k^g \in \mathbb{R}^{m_{k:g}\times r_g}$, $\mathbf{\Sigma}_k^g$ is $r_g \times r_g$ diagonal matrix, Ω_k^g is the index set of the observed samples of matrix $\mathbf{X}_k^g$. $m_{k:g}$ is the number of columns of $\mathbf{X}_k^g$. r_g is the matrix rank of $\mathbf{X}_k^g$.

The problem above can be solved by recovering each matrix independently. However, with the data correlation across days, a better recovery can be made if the set of matrices can be integrated into a tensor to recover together. This is not possible with each matrix having different number of columns. To address the issue, we first exploit the common data features hidden in the user domain and temporal domain to translate the problem.

As different monitoring matrices record the traffic data of the same set of n OD pairs of different days, they should share some common features in the user domain. Taking advantage of these features for more accurate traffic recovery, we use the same factor matrix $\mathbf{U}^g = \mathbf{U}_k^g$ (where $\mathbf{U}^g \in \mathbb{R}^{n\times r_g}$) for different sub-matrices of different days in Eq(8), and the transformed problem can be expressed as follows:

$$\begin{aligned}&\min_{\mathbf{U}^g,\mathbf{\Sigma}_k^g,\mathbf{V}_k^g} f\left(\mathbf{U}^g,\mathbf{\Sigma}_k^g,\mathbf{V}_k^g\right)\\ &\text{s.t. } f\left(\mathbf{U}^g,\mathbf{\Sigma}_k^g,\mathbf{V}_k^g\right)\\ &\qquad=\frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{X}_k^g-\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}_k^g)^T\right)_{\Omega_k^g}\right\|_F^2\right)\end{aligned} \tag{9}$$

Beside the common feature in user domain, as we have discussed in Section IV-D, traffic data also have common feature in the time domain, which is not captured in the problem (9). To reflect the feature, enlightened by Harshman [42], we impose an ***invariance constraint*** on the factor matrices $\mathbf{V}_k^g$ in the time domain: the cross product $(\mathbf{V}_k^g)^T\mathbf{V}_k^g$ is constant over different days, that is, $\mathbf{\Phi}^g = (\mathbf{V}_k^g)^T\mathbf{V}_k^g$ for $k = 1, 2, \cdots, K$.

Before we update the problem formulation in (9) to incorporate this invariance constraint, the following theorem reformulates the constraint.

Theorem 1: For the invariance constraint $\mathbf{\Phi}^g = (\mathbf{V}_k^g)^T\mathbf{V}_k^g$ to hold, it is necessary and sufficient to have $\mathbf{V}_k^g = \mathbf{P}_k^g\mathbf{V}^g$ where $\mathbf{V}^g \in \mathbb{R}^{r_g\times r_g}$ does not change in different days and $\mathbf{P}_k^g \in \mathbb{R}^{m_{k:g}\times r_g}$ is a column-wise orthonormal matrix with $(\mathbf{P}_k^g)^T\mathbf{P}_k^g = \mathbf{I}$.

Proof: The proof includes two parts.

1) ***Sufficiency proof.*** With $\mathbf{V}_k^g = \mathbf{P}_k^g\mathbf{V}^g$, $(\mathbf{V}_k^g)^T\mathbf{V}_k^g = (\mathbf{P}_k^g\mathbf{V}^g)^T\mathbf{P}_k^g\mathbf{V}^g = (\mathbf{V}^g)^T\mathbf{V}^g$ holds, that is, the invariance constraint is enforced.

2) ***Necessity proof.*** As $(\mathbf{V}_k^g)^T\mathbf{V}_k^g=(\mathbf{V}_j^g)^T\mathbf{V}_j^g$ for all pairs $j, k = 1, 2, \cdots, K$, we have $(\mathbf{V}_k^g)^T\mathbf{V}_k^g=(\mathbf{V}_1^g)^T\mathbf{V}_1^g$ for $k = 1, 2, \cdots, K$. We express $\mathbf{V}_k^g$ with respect to the column-wise orthogonal basis matrix $\mathbf{Q}_k^g \in \mathbb{R}^{m_{k:g}\times r_g}$ as $\mathbf{V}_k^g= \mathbf{Q}_k^g\mathbf{T}_k^g$ where $\mathbf{T}_k^g \in \mathbb{R}^{r_g\times r_g}$. Then it follows that $(\mathbf{T}_k^g)^T\mathbf{T}_k^g = (\mathbf{T}_1^g)^T\mathbf{T}_1^g$ and hence $\mathbf{T}_k^g=\mathbf{N}_k^g\mathbf{T}_1^g$, where $\mathbf{N}_k^g \in \mathbb{R}^{r_g\times r_g}$ is an orthonormal matrix. Therefore, $\mathbf{V}_k^g= \mathbf{Q}_k^g\mathbf{N}_k^g\mathbf{T}_1^g$ and we can define $\mathbf{P}_k^g= \mathbf{Q}_k^g\mathbf{N}_k^g$ and $\mathbf{V}^g=\mathbf{T}_1^g$. ■

Based on Theorem 1, the problem in (9) can be further transformed to (10) by replacing $\mathbf{V}_k^g$ with $\mathbf{P}_k^g\mathbf{V}^g$:

$$\begin{aligned}&\min_{\mathbf{U}^g,\mathbf{\Sigma}_k^g,\mathbf{V}^g,\mathbf{P}_k^g} f\left(\mathbf{U}^g,\mathbf{\Sigma}_k^g,\mathbf{V}^g,\mathbf{P}_k^g\right)\\ &\text{s.t. } f\left(\mathbf{U}^g,\mathbf{\Sigma}_k^g,\mathbf{V}^g,\mathbf{P}_k^g\right)\\ &\qquad=\frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{X}_k^g-\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^g\mathbf{V}^g)^T\right)_{\Omega_k^g}\right\|_F^2\right)\\ &\qquad(\mathbf{P}_k^g)^T\mathbf{P}_k^g=\mathbf{I}\end{aligned} \tag{10}$$

That is, the difference of the matrix $\mathbf{X}_k^g$ for different days $k = 1, 2, \cdots, K$ is captured by the matrix $\mathbf{\Sigma}_k^g$ and $\mathbf{P}_k^g$. In Section VI-B, we will show that the problem formulation in (10) provides the possibility of translating the matrix completion problem to the tensor completion through CP decomposition.

VI. Matrix Reshaping and Alignment for Tensor Completion

To solve the problem (10), we propose an alternating least squares algorithm that alternately solves the following two sub-problems:

- ***Sub-problem 1***: minimize (10) over $\mathbf{P}_k^g$ for a given set of $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$
- ***Sub-problem 2***: minimize (10) over $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$ for fixed $\mathbf{P}_k^g$

A. Sub Problem 1

The sub-problem 1 can be written as follows.

$$\begin{aligned}&\min_{\mathbf{P}_k^g} f\left(\mathbf{P}_k^g\right)\\ &\text{s.t. } f\left(\mathbf{P}_k^g\right)=\frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{X}_k^g-\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^g\mathbf{V}^g)^T\right)_{\Omega_k^g}\right\|_F^2\right)\\ &\qquad(\mathbf{P}_k^g)^T\mathbf{P}_k^g=\mathbf{I}\end{aligned} \tag{11}$$

Let

$$\mathbf{B} = \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^gV^g)^T, \tag{12}$$

we have $\mathbf{B}=\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T\left(\mathbf{P}_k^g\right)^T$ and $\mathbf{B}^T = \mathbf{P}_k^g\mathbf{V}^g\mathbf{\Sigma}_k^g(\mathbf{U}^g)^T$. The loss function on each sub-matrix (i.e. $\mathbf{X}_k^g$) can be written as

$$\begin{aligned}&\left\|\mathbf{X}_k^g-\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^g\mathbf{V}^g)^T\right\|_F^2\\ &\quad=tr\left(\left(\mathbf{X}_k^g-\mathbf{B}\right)\left(\mathbf{X}_k^g-\mathbf{B}\right)^T\right)\\ &\quad=tr\left(\left(\mathbf{X}_k^g-\mathbf{B}\right)\left(\left(\mathbf{X}_k^g\right)^T-\mathbf{B}^T\right)\right)\\ &\quad=tr\left(\mathbf{X}_k^g(\mathbf{X}_k^g)^T\right)-2tr\left(\mathbf{X}_k^g\mathbf{B}^T\right)+tr\left(\mathbf{B}\mathbf{B}^T\right)\end{aligned} \tag{13}$$

As $tr\left(\mathbf{X}_k^g(\mathbf{X}_k^g)^T\right)$ and $tr(\mathbf{B}\mathbf{B}^T) = tr(\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T\mathbf{V}^g\mathbf{\Sigma}_k^g(\mathbf{U}^g)^T)$ do not depend on $\mathbf{P}_k^g$, minimizing (10) is equivalent to solving the following problem:

$$\begin{aligned}&\max_{\mathbf{P}_k^g} tr\left(\mathbf{X}_k^g\mathbf{B}^T\right)\\ &\text{subject to: } (\mathbf{P}_k^g)^T\mathbf{P}_k^g=\mathbf{I}\\ &\qquad\mathbf{B}=\mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^gV^g)^T\end{aligned} \tag{14}$$

As $tr\left(\mathbf{X}_k^g \mathbf{B}^T\right) = tr\left(\mathbf{X}_k^g \mathbf{P}_k^g \mathbf{V}^g \Sigma_k^g (\mathbf{U}^g)^T\right) = tr\left(\mathbf{V}^g \mathbf{\Sigma}_k^g (\mathbf{U}^g)^T \mathbf{X}_k^g \mathbf{P}_k^g\right)$, the problem in (14) can be further transformed to

$$\max_{\mathbf{P}_k^g} tr\left(\mathbf{V}^g \mathbf{\Sigma}_k^g (\mathbf{U}^g)^T \mathbf{X}_k^g P_k^g\right)$$
$$\text{subject to: } (\mathbf{P}_k^g)^T \mathbf{P}_k^g = \mathbf{I} \tag{15}$$

Let $\mathbf{V}^g \mathbf{\Sigma}_k^g (\mathbf{U}^g)^T \mathbf{X}_k^g = \mathbf{M}_k^g \mathbf{\Delta}_k^g (\mathbf{N}_k^g)^T$ be singular value decomposition (SVD). According to [43], we have that $\mathbf{P}_k^g = \mathbf{N}_k^g (\mathbf{M}_k^g)^T$ is the column wise orthonormal solution for the problem (15).

B. Sub-Problem 2

Theorem 2: The sub-problem 2 of minimizing (10) over $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$ for fixed $\mathbf{P}_k^g$ can be reduced to the following problem:

$$\min_{\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g} f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right)$$
$$s.t.\ f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right) = \frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{X}_k^g \mathbf{P}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T\right)_{\mathbf{\Omega}_k^g}\right\|_F^2\right). \tag{16}$$

Proof: As we can easily see, the loss function in Eq(16) can be written as

$$\begin{aligned}
&\left\|\mathbf{X}_k^g \mathbf{P}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T\right\|_F^2 \\
&= \left((\mathbf{X}_k^g \mathbf{P}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T)(\mathbf{X}_k^g P_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T)^T\right) \\
&= \left(\mathbf{X}_k^g \mathbf{P}_k^g (\mathbf{X}_k^g \mathbf{P}_k^g)^T\right) - 2tr\left(\mathbf{X}_k^g \mathbf{P}_k^g (\mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T)^T\right) \\
&\quad + \left(\mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T (\mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T)^T\right) \\
&= \left(\mathbf{X}_k^g (\mathbf{X}_k^g)^T\right) - 2tr\left(\mathbf{X}_k^g \mathbf{P}_k^g \mathbf{V}^g (\mathbf{\Sigma}_k^g)^T (\mathbf{U}^g)^T\right) \\
&\quad + \left(\mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T \mathbf{V}^g (\mathbf{\Sigma}_k^g)^T (\mathbf{U}^g)^T\right)
\end{aligned} \tag{17}$$

Replacing $\mathbf{B}$ with $\mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{P}_k^g V^g)^T$ in Eq(12) to Eq.(17), we have

$$\left\|\mathbf{X}_k^g \mathbf{P}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T\right\|_F^2 = tr\left(\mathbf{X}_k^g (\mathbf{X}_k^g)^T\right) - 2tr\left(\mathbf{X}_k^g \mathbf{B}^T\right) + tr\left(\mathbf{B}\mathbf{B}^T\right) \tag{18}$$

Obviously, combining Eq.(13) and Eq.(18), we have

$$\left\|\mathbf{X}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{P}_k^g \mathbf{V}^g)^T\right\|_F^2 = \left\|\mathbf{X}_k^g \mathbf{P}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T\right\|_F^2, \tag{19}$$

the proof completes. ■

Let $\mathbf{Y}_k^g = \mathbf{X}_k^g \mathbf{P}_k^g$, problem in (16) can be further written as follows.

$$\min_{\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g} f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right)$$
$$\text{s.t. } f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right) = \frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{Y}_k^g - \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T\right)_{\mathbf{\Omega}_k^g}\right\|_F^2\right) \tag{20}$$

As $\mathbf{X}_k^g \in \mathbb{R}^{n \times m_{k:g}}$ and $\mathbf{P}_k^g \in \mathbb{R}^{m_{k:g} \times r_g}$, obviously, $\mathbf{Y}_k^g \in \mathbb{R}^{n \times r_g}$ has the identical size of $n \times r_g$. It is easy to see

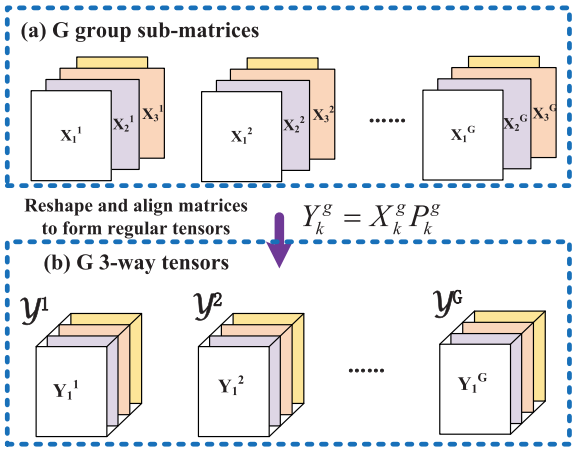


Fig. 9. Transform G groups of sub-matrices to G tensors. (a) G group sub-matrices. (b) G 3-way tensors.

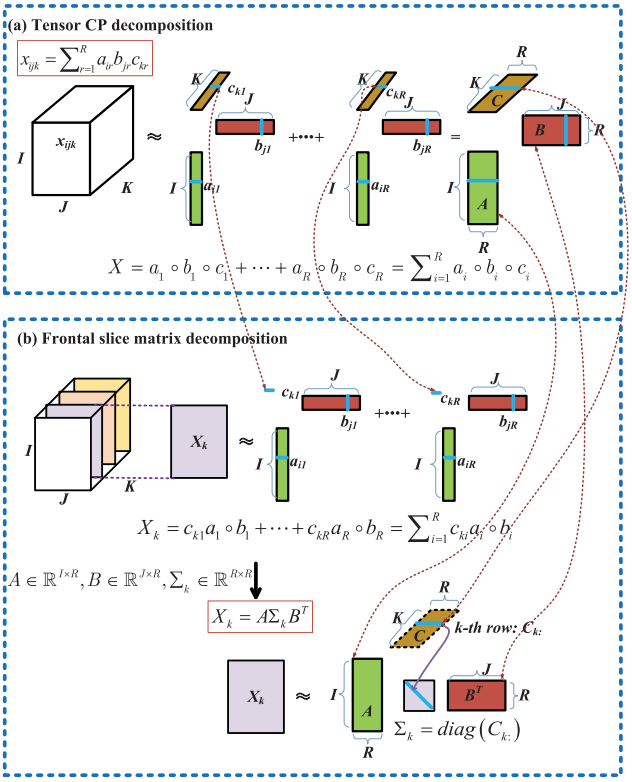


Fig. 10. The relationship between tensor CP decomposition and frontal slice matrix decomposition. (a) Tensor CP decomposition. (b) Frontal slice matrix decomposition.

in Fig. 9 (b), a group of K identical-size matrices $\mathbf{Y}_k^g$ for $1 \le k \le K$ forms a 3-way tensor with each slice being $\mathbf{Y}_k^g$. Therefore, G groups of identical-size matrices in Eq.(20) correspond to G 3-way tensors, as shown in Fig. 9.

Fig. 9 shows that multiple sub-matrices can be reshaped and aligned to the tensor-style. However, the problem in (20) is still a matrix completion problem. We would like to solve the problem through the tensor completion taking advantage of correlation across days for more accurate data recovery.

1) Relationship Between CP Decomposition and Frontal Slice Decomposition: To see if it is possible to translate the problem (20) to the tensor completion, we first investigate

the relationship between tensor CP decomposition and the decomposition of a frontal matrix slice of the tensor.

In Fig. 10(a), according to (5), the CP decomposition of a 3-way tensor $\mathcal{X}$ can be written as follows.

$$\mathcal{X} = \sum\nolimits_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{21}$$

where matrices $\mathbf{A} \in \mathbb{R}^{I\times R}$, $\mathbf{B} \in \mathbb{R}^{J\times R}$, $\mathbf{C} \in \mathbb{R}^{K\times R}$ are the factor matrices in the CP decomposition.

According to the CP decomposition, in Fig. 10(b), a frontal slice $\mathbf{X}_k$ can be written as

$$\mathbf{X}_k = c_{k1}\mathbf{a}_1 \circ \mathbf{b}_1 + \cdots + c_{kR}\mathbf{a}_R \circ \mathbf{b}_R = \sum\nolimits_{i=1}^{R} c_{ki}\mathbf{a}_i \circ \mathbf{b}_i \tag{22}$$

where $c_{k1}, c_{k2}, \cdots, c_{kR}$ are the entries of the k-th row of the factor matrix $\mathbf{C}$.

Eq (22) shows that each frontal slice $\mathbf{X}_k$ can be expressed as a superposition of R rank-1 matrices $\mathbf{a}_i \circ \mathbf{b}_i$ $(1 \le i \le R)$. That is, the traffic data $\mathbf{X}_k$ is approximated by the linear combination of R rank-1 matrices $\mathbf{a}_i \circ \mathbf{b}_i$. As $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_R] \in \mathbb{R}^{I\times R}$, $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_R] \in \mathbb{R}^{J\times R}$, according to (22), $\mathbf{X}_k$ can be rewritten as

$$\mathbf{X}_k = \mathbf{A}\mathbf{\Sigma}_k\mathbf{B}^T \tag{23}$$

where $\mathbf{A} \in \mathbb{R}^{I\times R}$ and $\mathbf{B} \in \mathbb{R}^{J\times R}$ are the factor matrices in the CP decomposition, $\mathbf{\Sigma}_k = diag\,(\mathbf{C}_{k:})$ and $\mathbf{C}_{k:}$ is the k-th row of the factor matrix $\mathbf{C}$.

2) Problem Transformation: The relationship shown in Fig. 10 provides a way to decompose a group of matrices with the same size through the tensor decomposition. The problem in (20) aims to find the matrix decomposition for matrix completion. The problem can be solved with higher accuracy if all the matrices can be integrated into a tensor. Fortunately, to capture the common user domain and time domain features, we have introduced the same user factor matrix $\mathbf{U}^g$ and time factor matrix $\mathbf{V}^g$ across different days. With the relationship between the tensor CP decomposition and the frontal matrix-slice decomposition, the formulation in (20) can be transformed to the following tensor completion problem:

$$\begin{aligned} &\min_{\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g} f\,(\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g) \\ &\text{s.t. } f\,(\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g) = \frac{1}{2}\sum_{g=1}^{G}\left(\|(\mathcal{Y}^g - [\![\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g]\!])_{\Omega^g}\|_F^2\right) \end{aligned} \tag{24}$$

where $\mathcal{Y}^g$ is the tensor with its frontal slices being sub-matrices $\mathbf{Y}_k^g$ for $k = 1, 2, \cdots, K$, $\mathbf{U}^g$,$\mathbf{V}^g$,$\mathbf{C}^g$ are the factor matrices of $\mathcal{Y}^g$, Ω^g is the index set of the observed samples of tensor $\mathcal{Y}^g$. As this paper dose not focus on CP decomposition methods, we apply the approach in [44] to perform the tensor completion. This allows us to obtain the optimal factor matrices $\mathbf{U}^g$,$\mathbf{V}^g$,$\mathbf{C}^g$ by minimizing the loss function in (24).

After obtaining $\mathbf{U}^g$,$\mathbf{V}^g$,$\mathbf{C}^g$, the reshaped slice can be recovered with $\hat{\mathbf{Y}}_k^g = \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T$ where $\mathbf{\Sigma}_k^g = diag\,(\mathbf{C}^g{}_{k:})$ and $\mathbf{C}^g{}_{k:}$ is the k-th row of the factor matrix $\mathbf{C}^g$. Then through the reverse procedure of reshaping, we can obtain the recovered sub traffic matrix $\hat{\mathbf{X}}_k^g = \hat{\mathbf{Y}}_k^g(\mathbf{P}_k^g)^T = \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T(\mathbf{P}_k^g)^T$.

VII. Complete Solution

In this section, we first present the complete solution, and then discuss the convergence of the solution.

A. Algorithm Design

The complete data recovery based on reshaping and alignment is shown in Algorithm 1. The sub-problems 1 and 2 are iteratively solved and 3-9 Steps are repeated until it converges.

Specially, given traffic matrices of K days, if there are $G-1$ time spots besides the time spots at 0:00 and 24:00 in the time rule involved in these K days, in Step 1, the large matrix of each day is divided into G sub-matrices according to the time alignment requirement. As there are K days, after such a division, there are G groups of sub-matrices with each group having K sub-matrices. The Step 2 initializes the factor matrices needed in the algorithm. Step 4 solves the sub problem 1 of minimizing (11) over $\mathbf{P}_k^g$ for fixed $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$. Step 5 builds the tensor with the reshaped sub-matrices. Step 6 solves the sub problem 2 and updates $\mathbf{U}^g$, $\mathbf{V}^g$, $\mathbf{C}^g$ by solving the tensor completion problem $\min_{\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g} f\,(\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g) = \frac{1}{2}\|(\mathcal{Y}^g - [\![\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g]\!])_{\Omega^g}\|_F^2$. Step 7 builds the diagonal matrix needed in the matrix decomposition $\mathbf{\Sigma}_k^g \leftarrow diag\,(\mathbf{C}_{k:}^g)$ where $\mathbf{C}_{k:}^g$ is the k-th row of factor matrix $\mathbf{C}^g$ obtained in Step 6. After obtaining $\mathbf{U}^g$, $\mathbf{V}^g$, $\mathbf{\Sigma}_k^g$, and $\mathbf{P}_k^g$, Step 8 calculates the recovered sub matrices in the iterative step.

In Step 8, $\mathbf{M}_k^g$ is an indicator matrix whose entry (i, j) is one if the entry (i, j) in $\mathbf{X}_k^g$ is sampled (i.e., measured) and zero otherwise. $\mathbf{1}$ is an all ones matrix that has the same size as $\mathbf{M}_k^g$. $\odot$ in Step 8 represents a scalar product (or dot product) of two matrices. For example, given that $\mathbf{A}$, $\mathbf{B}$ have the same size and $\mathbf{C} = \mathbf{A} \odot \mathbf{B}$, we have $c_{ij} = a_{ij}b_{ij}$. $\mathbf{X}_k^g \leftarrow \mathbf{M}_k^g \odot \mathbf{X}_k^g + (1 - \mathbf{M}_k^g) \odot \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T(\mathbf{P}_k^g)^T$ guarantees that the sample entry already measured remains unchanged and only the missing data are updated during the iterative procedure.

B. Convergence Analysis

The convergence of the proposed Algorithm 1 is guaranteed by the following theorem.

Theorem 3: The sequence $\{(\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t\}$ *generated by Algorithm 1 monotonically decreases the objective function* $f\,(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g, \mathbf{P}_k^g)$ *of Eq(10) where* t *denotes the iteration step; the objective function sequence* $f\,((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t)$ *converges; the sequence* $\{(\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t\}$ *converges.*

Proof: The proof includes three parts.

Part 1. We show that the update steps in step 4 and step 6 monotonically reduce the values in the objective function of Eq(10).

In step 4, given $(\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t$, we have $(\mathbf{P}_k^g)^{t+1} = \arg\min_{\mathbf{P}_k^g} f\,((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, \mathbf{P}_k^g)$. Therefore, we have

$$\begin{aligned} &f\left((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^{t+1}\right) \\ &\qquad \le f\left((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t\right). \end{aligned} \tag{25}$$

Similarly in step 6, given $(\mathbf{P}_k^g)^{t+1}$, we have $(\mathbf{U}^g)^{t+1}, (\mathbf{\Sigma}_k^g)^{t+1}, (\mathbf{V}^g)^{t+1} =$

Algorithm 1 Complete Reshape and Align Traffic Recovery Algorithm

According to the time alignment requirement, large matrices are divided into G groups of sub-matrices with each group having K sub matrices
Initialize $\mathbf{U}^g$ principal eigenvectors $\sum_{k=1}^{K} \mathbf{X}_k^g (\mathbf{X}_k^g)^T$ by SVD, $\mathbf{V}^g \leftarrow \mathbf{I}$, $\mathbf{\Sigma}_k^g \leftarrow \mathbf{I}$
while not converge **do**
  Sub problem 1: Compute the SVD $\mathbf{V}^g \Sigma_k^g (\mathbf{U}^g)^T \mathbf{X}_k^g = \mathbf{M}_k^g \mathbf{\Delta}_k^g (\mathbf{N}_k^g)^T$ and update $\mathbf{P}_k^g = \mathbf{N}_k^g (\mathbf{M}_k^g)^T$
  Generate tensor $\mathcal{Y}^g$ whose slices are $\mathbf{Y}_k^g = \mathbf{X}_k^g \mathbf{P}_k^g$
  Sub problem 2: Update $\mathbf{U}^g$, $\mathbf{V}^g$, $\mathbf{C}^g$ by solving $\min_{\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g} f(\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g) = \frac{1}{2} \| (\mathcal{Y}^g - [\![\mathbf{U}^g, \mathbf{V}^g, \mathbf{C}^g]\!])_{\Omega^g} \|_F^2$ for all the $g = 1, 2, \cdots, G$ tensors through CP decomposition
  $\mathbf{\Sigma}_k^g \leftarrow diag(\mathbf{C}_{k:}^g)$
  Update $\mathbf{X}_k^g \leftarrow \mathbf{M}_k^g \odot \mathbf{X}_k^g + (\mathbf{1} - \mathbf{M}_k^g) \odot \mathbf{U}^g \mathbf{\Sigma}_k^g (\mathbf{V}^g)^T (\mathbf{P}_k^g)^T$
end while
Combine the recovered sub-matrices and obtain the recovered large matrices.

$\arg\min_{\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g} f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g, (\mathbf{P}_k^g)^{t+1}\right)$ and thus

$$f\left((\mathbf{U}^g)^{t+1}, (\mathbf{\Sigma}_k^g)^{t+1}, (\mathbf{V}^g)^{t+1}, (\mathbf{P}_k^g)^{t+1}\right) \leq f\left((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^{t+1}\right). \quad (26)$$

Combining Eq(25) and Eq(26), we obtain

$$f\left((\mathbf{U}^g)^{t+1}, (\mathbf{\Sigma}_k^g)^{t+1}, (\mathbf{V}^g)^{t+1}, (\mathbf{P}_k^g)^{t+1}\right) \leq f\left((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t\right). \quad (27)$$

Part 2. We show that the objective function f is lower bounded. As $f(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g, \mathbf{P}_k^g)$ is the loss function defined based on the Frobenius norm $\|\|_F$, obviously, we have $f((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t) \geq 0$.

Thus, together with the monotonic decrease proved in Part 1, we can conclude that $f((\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t)$ converges.

Part 3. It is easy to see that $f(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g, \mathbf{P}_k^g)$ is Lipschitz continuous w.r.t $\mathbf{U}^g$, $\mathbf{\Sigma}_k^g$, $\mathbf{V}^g$, and $\mathbf{P}_k^g$, thus $\{(\mathbf{U}^g)^t, (\mathbf{\Sigma}_k^g)^t, (\mathbf{V}^g)^t, (\mathbf{P}_k^g)^t\}$ converges. ■

VIII. Performance Evaluations

In this section, we first present the simulation setting, and then the simulation results.

A. Simulation Setup

We use the public traffic trace data from two sources (the U. S. Internet2 Network (Abilene [40]) and the pan-European research backbone network (GÈANT [41])) to evaluate the performance of our proposed Reshape-Align scheme. The Abilene network consists of 12 nodes thus 144 OD pairs. The GÈANT network consists of 23 nodes thus 529 OD pairs. Abilene and GÈANT collect monitoring data every 5 minutes and 15 minutes respectively. Abilene and GÈANT record monitoring data in 168 days and 112 days, respectively.

TABLE I
Performance Metric

Error Ratio	$\frac{\sqrt{\sum_{k=1}^{K} \left(\sum_{(i,j)\in\bar{\Omega}_k} \left(x_{k:ij} - \hat{x}_{k:ij}\right)^2\right)}}{\sqrt{\sum_{k=1}^{K} \left(\sum_{(i,j)\in\bar{\Omega}_k} \left(x_{k:ij}\right)^2\right)}}$
MAE	$\frac{1}{\sum_{k=1}^{K} n \times m_k} \sum_{k=1}^{K} \left(\sum_{i,j} \lvert x_{k:ij} - \hat{x}_{k:ij} \rvert\right)$
RMSE	$\sqrt{\frac{1}{\sum_{k=1}^{K} n \times m_k} \sum_{k=1}^{K} \left(\sum_{i,j} \left(x_{k:ij} - \hat{x}_{k:ij}\right)^2\right)}$

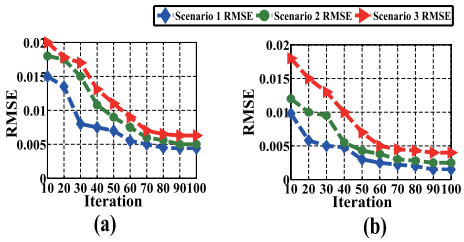


Fig. 11. Convergence behavior. (a) Abilene. (b) Geant.

Therefore, we have $K = 168$ and $K = 112$ for Abilene and GÈANT accordingly.

Three different metrics are considered: Error Ratio, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE), which are defined as Table I.

In the table, $x_{k:ij}$ and $\hat{x}_{k:ij}$ denote the raw data and the recovered data at (i,j)-th element of the matrix X_k where $1 \leq i \leq n, 1 \leq j \leq m_k$, and $1 \leq k \leq K$. Only entries not observed $(i,j) \in \bar{\Omega}_k$ are counted in the Error Ratio. Different from Error Ratio, the total data entries (i.e., $\sum_{k=1}^{K} n \times m_k$) are counted in the MAE and RMSE. MAE is an average of the absolute errors after the interpolation, RMSE is the standard deviation of the differences between recovered values and raw values.

According to the time alignment requirement in Section V, different measurement rates will result in different partitions. We take 3 measurement scenarios as examples to show the performance: 1) The measurement rates are different in different days while the measurement rate of the same day is the same. 2) The measurement rates are different in different days while the measurement rate changes at the noon every day. 3) The measurement rates are different in different days while the measurement rate changes at 8:00, and 16:00 every day. Obviously, for time alignment, matrices in Scenario 1 form one group. In Scenarios 2 and 3, the traffic matrices are partitioned into two groups and three groups, respectively.

B. Convergence Behavior

As presented in Section VI, two sub-problems are iteratively solved by an alternating least squares algorithm in our Reshape-Align scheme. Fig. 11 shows the convergence behavior of Reshape-Align where the sampling ratio is set to be 50%. Different from measurement rate that denotes how often measurements are taken in the network, the sampling ratio in the simulation is the fraction of the matrix entries that are observed with the measurements. In this work, the remaining entries are inferred with tensor completion in our Reshape-Align.

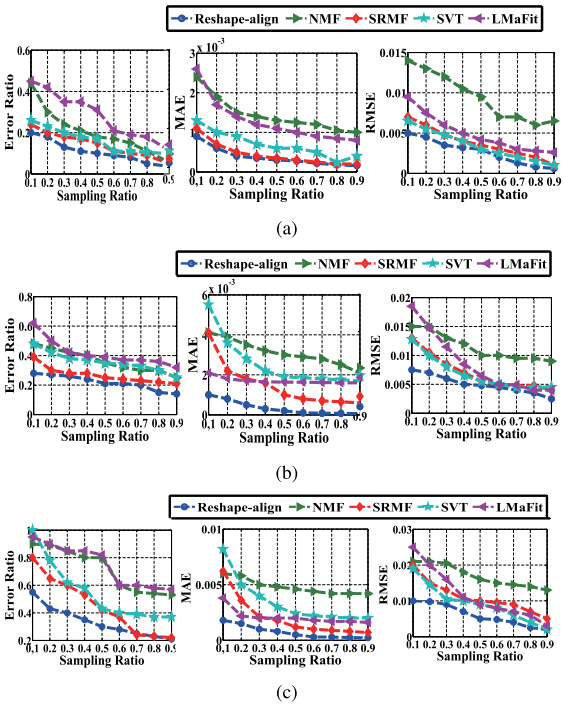


Fig. 12. Comparison with matrix completion algorithms using trace Abilene. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

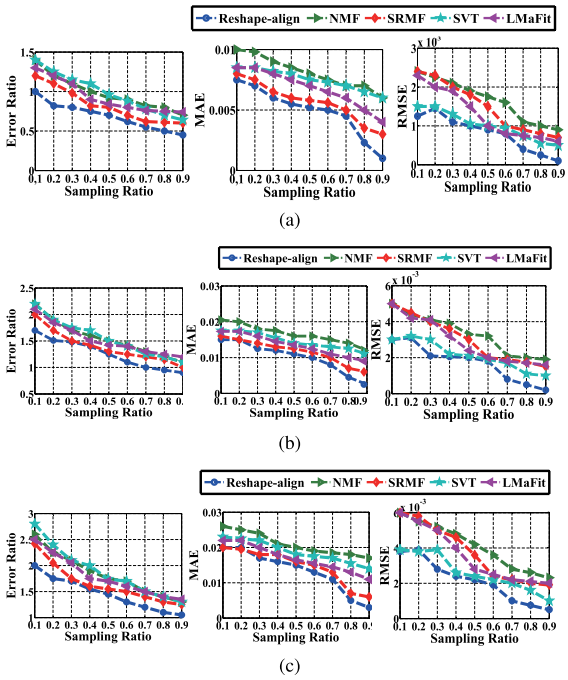


Fig. 13. Comparison with matrix completion algorithms using trace GÈANT. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

Obviously, in all the simulation scenarios using different trace data, RMSE decreases with iterations and converges to a stable value quickly. This demonstrates that it is efficient and effective to solve the traffic data recovery problem through the alternating least squares algorithm.

Among all the 3 scenarios, the Scenario 1 achieves the best recovery performance while Scenario 3 achieves the worst performance. This is because their matrix sizes are different. The matrices in the Scenario 1 cover the time segment of the whole day, while the sub-matrices for Scenario 2 correspond to half a day, and the sub-matrices operated in Scenario 3 cover one third of a day. A longer time period makes more data available to abstract the temporal feature for missing data recovery, and thus the best performance is achieved in Scenario 1.

C. Comparison With Matrix Completion Algorithms

Although some limited very recent studies consider the traffic data recovery through tensor completion, they cannot be directly applied in the practical network with variable measurement rates. Therefore, we implement four matrix completion algorithms (*NMF* [45], *SRMF* [11], *SV T* [46], *LMaFit* [47]) for the performance comparison.

To align measurement data under different measurement rates for data recovery, in all the above matrix completion algorithms, our temporal division scheme is taken to form the sub-matrices of each day. Then we combine the recovery results of different days to evaluate the performance.

1) Performance Under Different Sampling Ratios and Scenarios: From Fig. 12 to Fig. 13, we study the impact of sampling ratio under three different scenarios we set in the simulations. Fig. 12 (Abilene) and Fig. 13 (GÈANT) show the performance in terms of error ratio, MAE, and RMSE with different sampling ratios under the uniform sampling. The performance results are different for different traces. As expected, with the increase of the sampling ratio thus sample data, the error ratio, MAE, and RMSE decrease and thus better recovery performance is obtained. Our Reshape-Align scheme can transform the data recovery through the traffic matrix to the tensor completion to well exploit the multi-dimensional correlations hidden in the traffic data. Therefore, compared with other four matrix completion algorithm, Reshape-Align achieves the best recovery performance with the lowest error ratio, MAE, and RMSE in all the figures using different trace data. Among all the algorithms that directly use the matrix completion, SRMF achieves the best performance. Besides using a low-rank matrix to approximate the traffic matrix, SRMF also utilizes spatial and temporal constraint matrices in the problem formulation to express the knowledge about the spatio-temporal structure of the traffic matrix.

Similar to the results in Fig. 11, among all the scenarios, scenario 1 achieves the best performance while scenario 3 achieves the worst performance because more data are available to abstract the temporal feature to infer the missing data in scenario 1.

2) Performance Under Consecutive Data Missing: We randomly select one day and let consecutive measurements over 60 minutes all get lost in Scenario 3, and then calculate the error ratio on the 60 minute data, as shown in Fig. 14. The consecutive data missing, obviously, results in the consecutive column missing in the traffic matrix. From the literature work, the conventional matrix completion algorithms fail if there are completely empty rows or columns as they do not have effect on these missing entries. In the Fig. 14, we use N/A to indicate that these algorithms fail to function. Reshape-Align, however, can recover the consecutively missing data with the error ratio only 0.4 (Abilene).

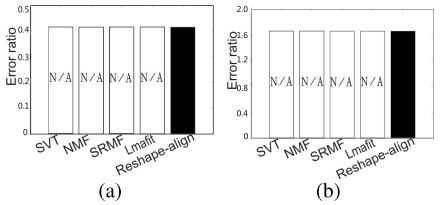


Fig. 14. Performance under consecutive data missing. (a) Abilene. (b) GÈANT.

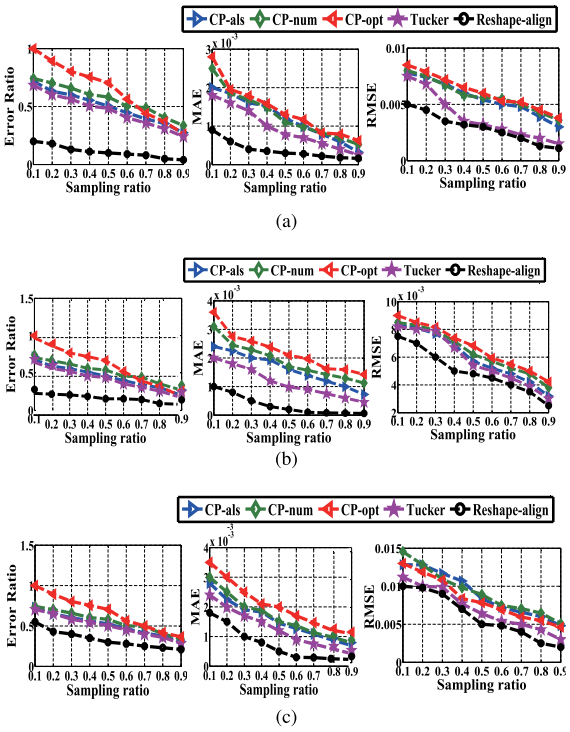


Fig. 15. Comparison with tensor completion algorithms using trace Abilene. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

With the time alignment and matrix reshaping, our Reshape-Align scheme transforms the problem of recovering the traffic data measured with variable rates into a tensor completion problem, taking advantage of the information along three dimensions to recover the missing data. The normal matrix completion only considers the constraints along two particular dimensions. This is the key reason why Reshape-Align outperforms the matrix completion-based algorithms.

D. Comparison With Tensor Completion Algorithms

Under variable measurement rates, the matrices of different days can not be directly combined as a regular tensor thus it is hard to apply current tensor completion algorithms to infer the missing data. To allow the tensor completion to be used, a trivial way can be applied to handle such an issue with four steps. First, our temporal division scheme can be applied to the traffic matrices of multiple days to align the time, with which multiple sub-matrices are created and form different groups. Second, for each group, a regular sub-tensor can be built by combining sub-matrices from different days with the lowest common multiple of time-stamps of all sub-matrices in the group as the dimensionality of the temporal

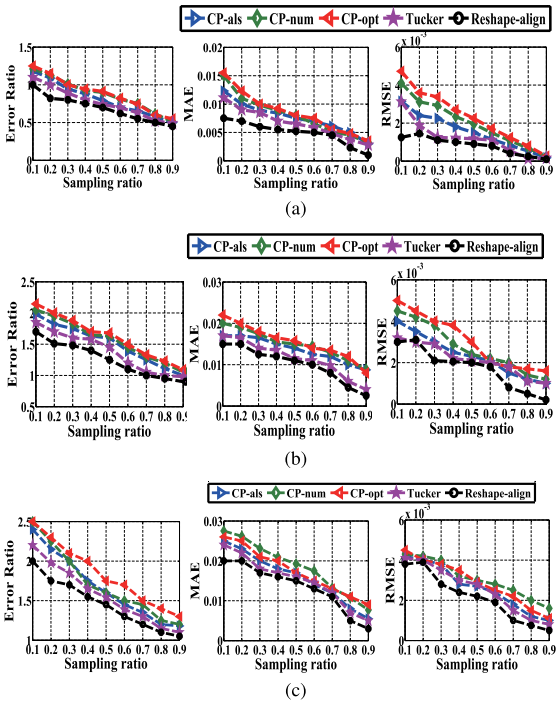


Fig. 16. Comparison with tensor completion algorithms using trace GÈANT. (a) Scenario 1. (b) Scenario 2. (c) Scenario 3.

domain. Third, a traditional tensor completion algorithm is applied to each sub-tensor to infer the missing data. Fourth, the recovery results of different groups are integrated to form the complete tensor. We apply this scheme to four tensor completion algorithms (*CPnmu* [33], *CPopt* [34], *CPals* [48], and *Tucker* [48]), and compare their results with those achieved with our Reshapealign.

The first three (CP_{nmu}, CP_{opt}, and CP_{als}) are designed based on CP model, the last $Tucker$ is designed based on the Tucker model.

As expected, with the increase of sampling ratio thus the number of measurements, the recovery performance under all tensor completion algorithms increase with the decrease of error ratio, MAE, and RMSE in Fig. 15 and Fig. 16. Among all the tensor completion algorithms implemented, our Reshape-Align achieves the best recovery performance with the lowest recovery error ratio, MAE, and RMSE. Under the trivial way of aligning the measurement data, the lowest common multiple of time-stamps of all sub-matrices is set as the dimensionality of the temporal domain. As a result, many columns in the matrices are empty, which makes the tensor very sparse and reduces the tensor completion performance. The results demonstrate that our technique on reshaping and aligning un-regular tensor to a regular tensor is very effective and efficient. Among all the algorithms that take the trivial method to build the tensor, $Tucker$ achieves the best performance as it utilizes a core tensor to coordinate the information along different tensor modes.

E. The Performance Impacted by the Number of Groups

To investigate how the number of groups G impacts the recovery performance, we vary G from 1 to 5. As shown in Section VIII-A, in our simulation scenarios 1, 2, and 3,

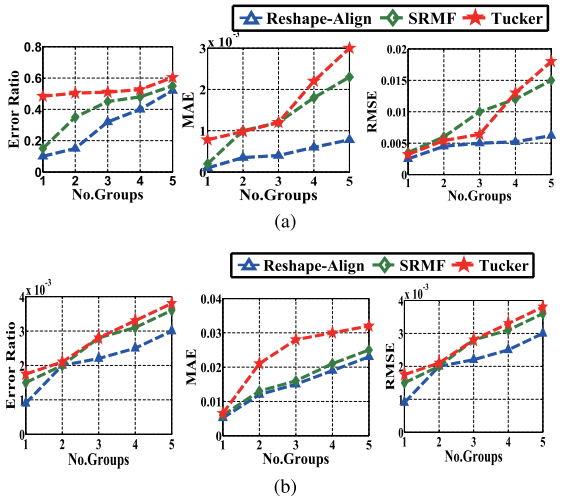


Fig. 17. Performance impacted by the number of groups (No.Groups). (a) Abilene. (b) GÈANT.

the traffic matrices are partitioned into one group, two groups and three groups, respectively. Following the simulation setting in Section VIII-A, scenarios 4 and 5 partition the traffic matrices into 4 and 5 groups, respectively.

Fig. 17 shows the recovery performance. As expected, with the increase of G, the recovery performance decreases with higher error ratio, MAE, and RMSE, as the relation from fewer data can be exploited in each group for inference. However, compared with SRMF (the best matrix completion algorithm) and Tucker (the best tensor completion algorithm), our Reshape-align can achieve better recovery performance, which demonstrates that our Reshape-align is more effective than existing matrix completion and tensor completion algorithms. The performance of the tensor completion algorithm Tucker is even worse than that of SRMF, as the trivial way of building the regular tensor will make the tensor very sparse and consequently reduce the recovery performance.

IX. Conclusion

Accurate inference of the traffic matrix in the presence of changing measurement frequency is of practical importance. In this paper, we propose a Reshape-Align scheme which can reshape the inconsistent traffic matrices observed in different days into consistent ones, align and integrate these matrices to form tensor, and take advantage of the user-domain and temporal domain features hidden in the traffic data to translate the matrix completion problem to the tensor completion problem with CP decomposition for more accurate missing data recovery. The performance studies demonstrate that, compared with the state of art matrix-completion and tensor completion algorithms, our scheme can infer missing data with significantly better performance.

References

[1] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: Making use of multiple data sources," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 133–144, 2006.

[2] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large IP backbone: A comparison on real data," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, 2004, pp. 149–160.

[3] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 206–217, 2003.

[4] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 161–174, 2002.

[5] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for OpenFlow networks," in *Proc. Int. Conf. Passive Active Netw. Meas.*, 2010, pp. 201–210.

[6] A. Lakhina *et al.*, "Structural analysis of network traffic flows," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 61–72, 2004.

[7] 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.

[8] 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.

[9] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. Amer. Statist. Assoc.*, vol. 91, no. 433, pp. 365–377, 1996.

[10] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, 2002, pp. 71–82.

[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] M. Mardani and G. B. Giannakis, "Robust network traffic estimation via sparsity and low rank," in *Proc. IEEE ICASSP*, May 2013, pp. 4529–4533.

[13] R. Du, C. Chen, B. Yang, and X. Guan, "VANET based traffic estimation: A matrix completion approach," in *Proc. IEEE GLOBECOM*, Dec. 2013, pp. 30–35.

[14] G. Gürsun and M. Crovella, "On traffic matrix completion in the internet," in *Proc. Internet Meas. Conf.*, 2012, pp. 399–412.

[15] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 545–556.

[16] K. Xie *et al.*, "Accurate recovery of Internet traffic data: A tensor completion approach," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[17] T. Kolda and B. Bader, "The tophits model for higher-order Web link analysis," in *Proc. Workshop Link Anal., Counterterrorism Secur.*, 2006, pp. 26–29.

[18] A. Carlson *et al.*, "Toward an architecture for never-ending language learning," in *Proc. AAAI*, 2010, pp. 1306–1313.

[19] C. J. Appellof and E. Davidson, "Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents," *Anal. Chem.*, vol. 53, no. 13, pp. 2053–2056, 1981.

[20] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[21] H. Tan *et al.*, "A tensor-based method for missing traffic data completion," *Transp. Res. C, Emerg. Technol.*, vol. 28, pp. 15–27, Mar. 2013.

[22] Y. Han and F. Moutarde, "Analysis of large-scale traffic dynamics in an urban transportation network using non-negative tensor factorization," *Int. J. Intell. Transp. Syst. Res.*, vol. 14, no. 1, pp. 36–49, 2016.

[23] S. Aja-Fernández, R. de Luis Garcia, D. Tao, and X. Li, *Tensors in Image Processing and Computer Vision*. London, U.K.: Springer, 2009.

[24] H. Zhou, D. Zhang, K. Xie, and Y. Chen, "Spatio-temporal tensor completion for imputing missing Internet traffic data," in *Proc. IEEE IPCCC*, Dec. 2015, pp. 1–7.

[25] K. Xie *et al.*, "Fast tensor factorization for accurate Internet anomaly detection," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3794–3807, Dec. 2017, doi: 10.1109/TNET.2017.2761704.

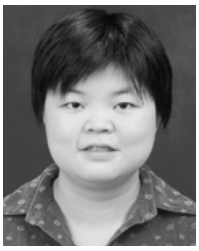
[26] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[27] R. A. Harshman, *Foundations of the PARAFAC Procedure: Models and Conditions for an 'Explanatory' Multimodal Factor Analysis*. Los Angeles, CA, USA: Univ. California, Los Angeles, 1970.

[28] K. Xie *et al.*, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Proc. IEEE INFOCOM*, 2015, pp. 2443–2451.

[29] J. Cheng, Y. Liu, Q. Ye, H. Du, and A. V. Vasilakos, "DISCS: A distributed coordinate system based on robust nonnegative matrix completion," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 934–947, Apr. 2017.

[30] Y. Liao, W. Du, P. Geurts, and G. Leduc, "DMFSGD: A decentralized matrix factorization algorithm for network distance prediction," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1511–1524, Oct. 2013.
[31] R. Zhu, B. Liu, D. Niu, Z. Li, and H. V. Zhao, "Network latency estimation for personal devices: A matrix completion approach," *IEEE/ACM Trans. Netw.*, vol. 25, no. 2, pp. 724–737, Apr. 2017.
[32] 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.
[33] 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.
[34] E. Acar, D. M. Dunlavy, and T. G. Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *J. Chemometrics*, vol. 25, no. 2, pp. 67–86, 2011.
[35] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.
[36] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
[37] H. Zhou, D. Zhang, K. Xie, and Y. Chen, "Robust spatio-temporal tensor recovery for Internet traffic data," in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 1404–1411.
[38] H. Xiao, J. Gao, D. S. Turaga, L. H. Vu, and A. Biem, "Temporal multiview inconsistency detection for network traffic analysis," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 455–465.
[39] I. Markovsky, *Low-Rank Approximation: Algorithms, Implementation, Applications*. London, U.K.: Springer, 2011.
[40] *The Abilene Observatory Data Collections*. Accessed: Jul. 20, 2004. [Online]. Available: http://abilene.internet2.edu/observatory/data-collections.html
[41] 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.
[42] R. A. Harshman, "Parafac2: Mathematical and technical notes," *UCLA Working Papers Phonetics*, vol. 22, no. 3044, p. 122215, 1972.
[43] B. F. Green, "The orthogonal approximation of an oblique structure in factor analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, 1952.
[44] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[45] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
[46] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[47] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.
[48] B. W. Bader *et al.*. (Jan. 2012). *MATLAB Tensor Toolbox Version 2.5*. [Online]. Available: http://www.sandia.gov/ tgkolda/TensorToolbox/



Kun Xie received the Ph.D. degree in computer application from Hunan University, Changsha, China, in 2007. She is currently a Professor with Hunan University. Her research interests include network monitoring and management, wireless network, mobile computing, and matrix and tensor factorization.

Can Peng is currently pursuing the master's degree with the College of Computer Science and Electronics Engineering, Hunan University. Her research interests include tensor completion and traffic data analysis.

Xin Wang (M'01) received the Ph.D. degree in electrical and computer engineering from Columbia University, New York, NY, USA. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, The State University of New York at Stony Brook, Stony Brook, NY, USA. Her research interests include algorithm and protocol design in wireless networks and communications, mobile and distributed computing, and networked sensing and detection. She received the NSF CAREER Award in 2005 and the ONR Challenge Award in 2010.

Gaogang Xie received the B.S. degree in physics and the M.S. and Ph.D. degrees in computer science from Hunan University in 1996, 1999, and 2002, respectively. He is currently a Professor and the Director of the Network Technology Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include Internet architecture, packet processing and forwarding, and Internet measurement.

Jigang Wen received the Ph.D. degree in computer application from Hunan University, China, in 2011. He was a Research Assistant with the Department of Computing, Hong Kong Polytechnic University, from 2008 to 2010. He is currently a Post-Doctoral Fellow with the Institute of Computing Technology, Chinese Academy of Science, China. His research interests include wireless network and mobile computing and high-speed network measurement and management.

Jiannong Cao (M'93–SM'05–F'14) received the Ph.D. degree in computer science from Washington State University, Pullman, WA, USA, in 1990. He is currently a Chair Professor and the Head of the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware.

Dafang Zhang received the Ph.D. degree in application mathematics from Hunan University, Changsha, China, in 1997. He is currently a Professor with Hunan University. His research interests include packet processing, Internet measurement, wireless network and mobile computing, and big data.

Zheng Qin received the Ph.D. degree in computer science from Chongqing University, China, in 2001. He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University, China. His main research interests include computer networking, network and information security, big data, and cloud computing.