Accurate Recovery of Internet Traffic Data Under Dynamic Measurements

Kun Xie[1,2], Can Peng[1], Xin Wang[2], Gaogang Xie[3], Jigang Wen[3]
[1] College of Computer Science and Electronics Engineering, Hunan University, Changsha, China
[2] Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA
[3] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
xiekun@hnu.edu.cn, pengcancaroline@gmail.com, x.wang@stonybrook.edu,
xie@ict.ac.cn, wenjigang@ict.ac.cn

***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 three-dimensional tensor as compared to interpolation methods based on two-dimensional 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 Reshape-Align scheme to form the regular tensor with data from dynamic measurements, and introduce user-domain and temporal-domain factor matrices which takes full advantage of features from both domains to translate the matrix completion problem to the tensor completion problem based on CP decomposition for more accurate missing data recovery. Our performance results demonstrate that our Reshape-Align scheme can achieve significantly better performance in terms of two metrics: error ratio and mean absolute error (MAE).**

***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]–[17]. Although the performance is good when the data missing ratio is low, the performance suffers when the missing ratio is large.

Based on the analyses of real traffic trace, recent work in [18] 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. The authors [18] thus model the traffic data as a 3-way tensor to concurrently consider the traffic of different days for more accurate missing data

The work is supported by the National Natural Science Foundation of China under Grant Nos.61572184, 61472130, and 61472131, and U.S. NSF CNS 1526843.

interpolation.

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 [19], knowledge bases [20], chemometrics [21], signal processing [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 [18], [24] is built with a strong assumption that the network monitoring system adopts a static measurement strategy with a fixed sampling 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 dynamic 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 dynamic measurements 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 CP decomposition.
- We compare the proposed Reshape-Align scheme with the state of the art matrix-based algorithms, and our results demonstrate that our scheme can achieve significantly better performance in terms of two metrics: error ratio and mean absolute error (MAE).

To the best of our knowledge, our Reshape-Align scheme is the first one that considers the traffic recovery problem under dynamic measurement 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 traffic matrix recovery algorithms [12]–[15], [17] are proposed to recover the missing data from partial traffic 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 is still low.

To further utilize the traffic periodicity feature for accurate traffic data recovery, the recent studies [18], [24] combine the traffic matrices of different days to form a tensor to recover the missing data. Several tensor completion algorithms [25]–[28] 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 [29], [30] and Tucker decomposition [31]). Tensor has proven to be

good data structure for dealing with the multi-dimensional data in a variety of fields [19]–[23]. Although promising, the traffic tensor model in [18], [24] is built with a strong assumption that the network monitoring system adopts a static measurement strategy with a fixed sampling rate. The proposed methods may fail to work in a practical network monitoring scenario where the rate of measurements varies over time.

Moreover, several novel techniques are proposed in the scheme such as matrix division algorithm for time alignment, mechanism to reshape and align matrices, and the technique to solve the matrix completion problem through tensor CP decomposition.

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 two metrics: error ratio and mean absolute error (MAE).

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.*

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

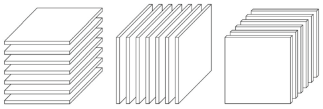


Fig. 1. Tensor slices

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 $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N_2}}$ is the tensor of the order $N_1 + N_2$ defined by*

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

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 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.

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

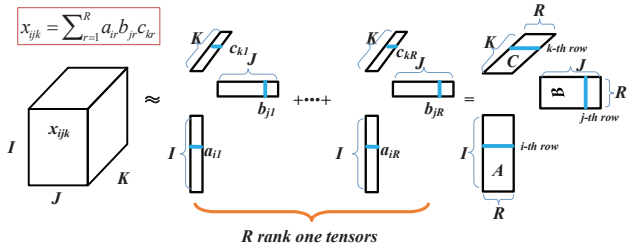


Fig. 2. CP decomposition of three-way tensor as sum of R outer products (rank one tensors). CP decomposition can be written as a triplet of factor matrices $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, i.e, the r-th column of which contains $\mathbf{a}_r$, $\mathbf{b}_r$, and $\mathbf{c}_r$, respectively.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

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

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

A. Traffic recovery problem based on matrix factorization

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.

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 K days. Specifically, to recover the missing data, a monitoring matrix $\mathbf{X}_k$ is 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 a $m_k \times r$ factor matrix $\mathbf{V}_k$ *for time domain* under the condition of minimizing the loss function as follows:

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

where r denotes the matrix rank, $f\left(\mathbf{U}_k, \mathbf{\Sigma}_k, \mathbf{V}_k\right) = \frac{1}{2} \sum_{k=1}^{K} \left\| \left(\mathbf{X}_k - \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T\right)_{\Omega_k} \right\|_F^2$ is the loss function defined based on the Frobenius norm $\|\|_F$, Ω_k is the index set of the observed samples of the matrix $\mathbf{X}_k$.

After obtaining the factor matrices $\mathbf{U}_k$, 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.

B. From matrix factorization to tensor factorization

As traffic data are observed to possess the features of temporal stability, spatial correlation, and periodicity features [18], [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 taking advantage of the periodicity feature of traffic across days. Despite the potential, in a practical network monitoring system, the measurement strategy may often vary according to the traffic conditions. There exist some challenges to combine multiple matrices to form a tensor:

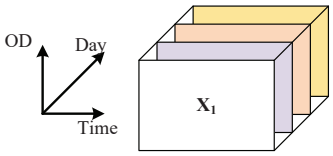
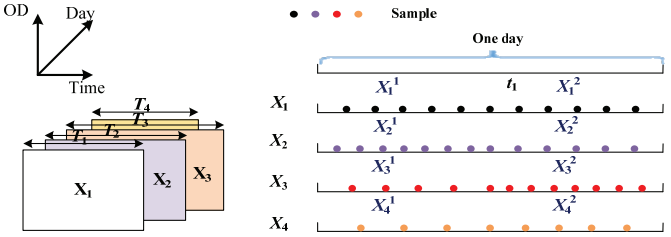


Fig. 3. Tensor based traffic model

- ***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.4(a)). This introduces the challenge to forming the standard tensor with these matrices.
- ***Inconsistent length of time slot within the matrix***. Different sampling rate makes columns in a matrix to correspond to different time-slot lengths (as shown in Fig.4(b)), which further brings the difficulty of recovering the missing items through the temporal and spatial correlation among traffic data



(a) Inconsistent number of columns (b) Different length of the time slot in matrices

Fig. 4. Traffic matrices with inconsistent number of columns

C. Characteristics in multiple data matrices

Although the variation of the sampling 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 (8), we use the same factor matrix $\mathbf{U}^g$ for different traffic matrices.
- 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.

The aims of this paper are to investigate and take advantage of the common features in the traffic matrices to reshape and align traffic matrices with inconsistent number of columns and time-slot lengths to form regular tensors, and design efficient tensor-based traffic data recovery algorithms for more accurate data recovery.

D. 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 frequency.

Fig. 5(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.5 (b), where the matrices are divided in temporal

domain to satisfy the time alignment requirement. The sub-matrices formed after the division (in Fig.5 (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 and temporal domain features hidden in the sub-matrices, and then integrate the reshaped and aligned sub-matrices to form the tensor in Fig. 5 (d). We apply the tensor completion algorithm to interpolate the missing data in Fig. 5 (e), and then take the reverse procedure of reshaping to obtain the recovered sub-matrices (in Fig. 5 (f)), which will be combined to form the final recovered large matrices (in Fig. 5 (g)).

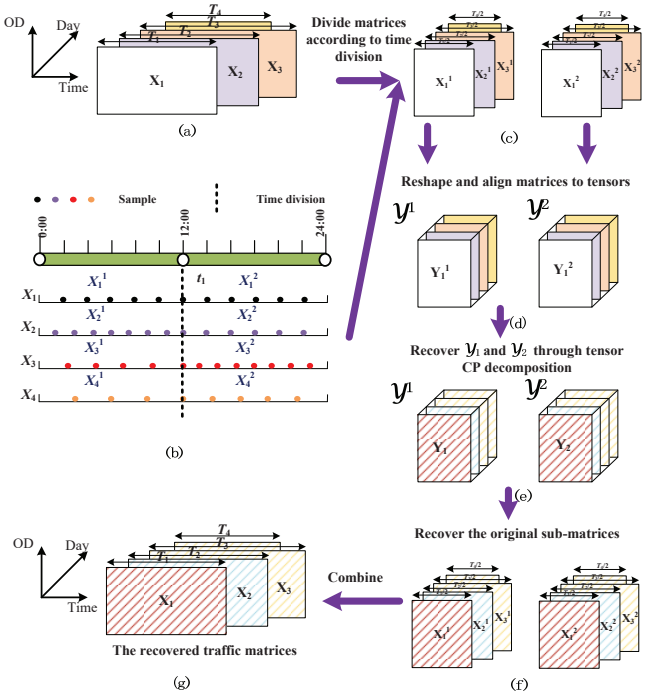


Fig. 5. Overview solution of Reshape-Align scheme

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 sampling rate.

Fig. 6 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. 6(a), a fixed measurement

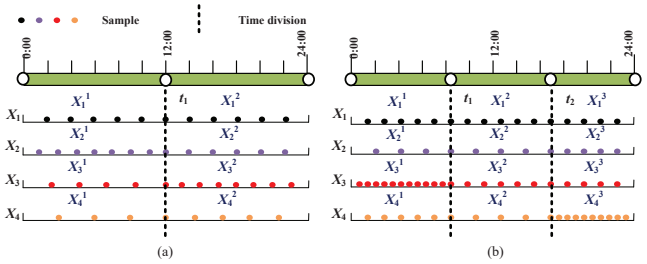


Fig. 6. Time alignment problem

rate is adopted in $\mathbf{X}_1$ for the whole day. For matrices $\mathbf{X}_2$, $\mathbf{X}_3$, $\mathbf{X}_4$, two different sampling 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.6(b), the time in the day is divided into three time segments and thus original traffic matrices are divided into three parts with $\mathbf{X}_1 = \left[\mathbf{X}_1^1, \mathbf{X}_1^2, \mathbf{X}_1^3\right]$, $\mathbf{X}_2 = \left[\mathbf{X}_2^1, \mathbf{X}_2^2, \mathbf{X}_2^3\right]$, $\mathbf{X}_3 = \left[\mathbf{X}_3^1, \mathbf{X}_3^2, \mathbf{X}_3^3\right]$, $\mathbf{X}_4 = \left[\mathbf{X}_4^1, \mathbf{X}_4^2, \mathbf{X}_4^3\right]$.

As shown in Fig.6, 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.

B. 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 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 timespot 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 loss function on each sub-matrices.

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

where $\mathbf{X}_k^g \in \mathbb{R}^{n \times m_{k:g}}$, $\mathbf{U}^g \in \mathbb{R}^{n \times r_g}$, $\mathbf{V}_k^g \in \mathbb{R}^{m_{k:g} \times r_g}$, $\mathbf{\Sigma}_k^g$ 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 $\mathbf{U}^g$ for different sub-matrices of different days in Eq(8).

Beside the common feature in user domain, as we have discussed in Section IV-C, traffic data also have common feature in the time domain, which is not captured in the problem (8). To reflect the feature, enlightened by Harshman [32], 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 (8) to incorporate this invariance constraint, the following theorem reformulates the constraint.

Theorem 1. *For the invariance constraint $(\mathbf{V}_k^g)^T\mathbf{V}_k^g$ 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 $(\mathbf{P}_k^g)^T\mathbf{P}_k^g = \mathbf{I}$.*

Due to the limited space, the proof is omitted.

To exploit the common feature in time domain, based on Theorem 1, replace $\mathbf{V}_k^g = \mathbf{P}_k^g\mathbf{V}^g$, the problem in (8) can be further transformed as follows:

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

That is, the difference of the matrix $\mathbf{X}_k^g$ for different days $k = 1, 2, \cdots, K$ are 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 (9) 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 (9), we propose an alternating least squares algorithm that alternately solves the following two sub-problems:

- ***Sub-problem 1***: minimize (9) over $\mathbf{P}_k^g$ for a given set of $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$
- ***Sub-problem 2***: minimize (9) 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.

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

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

$$\begin{aligned}&\left\|\mathbf{X}_k^g - \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^g\mathbf{V}^g)^T\right\|_F^2 = tr\left(\left(\mathbf{X}_k^g - \mathbf{B}\right)\left(\mathbf{X}_k^g - \mathbf{B}\right)^T\right)\\ &= tr\left(\left(\mathbf{X}_k^g - \mathbf{B}\right)\left(\left(\mathbf{X}_k^g\right)^T - \mathbf{B}^T\right)\right)\\ &= tr\left(\mathbf{X}_k^g(\mathbf{X}_k^g)^T\right) - 2tr\left(\mathbf{X}_k^g\mathbf{B}^T\right) + tr\left(\mathbf{B}\mathbf{B}^T\right)\end{aligned} \tag{11}$$

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 (9) is equal to solving the following problem:

$$\begin{aligned}&\max_{\mathbf{P}_k^g} tr\left(\mathbf{X}_k^g\mathbf{B}^T\right)\\ &\text{s.t. } \left(\mathbf{P}_k^g\right)^T\mathbf{P}_k^g = \mathbf{I}\\ &\mathbf{B} = \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{P}_k^gV^g)^T\end{aligned} \tag{12}$$

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 (12) can be further transformed to

$$\begin{aligned}&\max_{\mathbf{P}_k^g} tr\left(\mathbf{V}^g\mathbf{\Sigma}_k^g(\mathbf{U}^g)^T\mathbf{X}_k^gP_k^g\right)\\ &\text{s.t. } \left(\mathbf{P}_k^g\right)^T\mathbf{P}_k^g = I\end{aligned} \tag{13}$$

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 [33], 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 (13).

B. Sub-problem 2

The sub-problem 2 of minimizing (9) over $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$ for fixed $\mathbf{P}_k^g$ reduces to the following problem.

$$\begin{aligned}&\min_{\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g} f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right)\\ &\text{s.t. } f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right) = \frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{X}_k^g\mathbf{P}_k^g - \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T\right)_{\mathbf{\Omega}_k^g}\right\|_F^2\right)\end{aligned} \tag{14}$$

Let $\mathbf{Y}_k^g = \mathbf{X}_k^g\mathbf{P}_k^g$, problem in (14) can be further written as follows.

$$\begin{aligned}&\min_{\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g} f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right)\\ &\text{s.t. } f\left(\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g\right) = \frac{1}{2}\sum_{g=1}^{G}\left(\sum_{k=1}^{K}\left\|\left(\mathbf{Y}_k^g - \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T\right)_{\mathbf{\Omega}_k^g}\right\|_F^2\right)\end{aligned} \tag{15}$$

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, $\mathbf{Y}_k^g \in \mathbb{R}^{n \times r_g}$. It is easy to see Eq.(15) corresponds to G 3-way tensors as illustrated in Fig.7(b), where each slice $\mathbf{Y}_k^g$ has the identical size of $n \times r_g$.

Fig.7 shows that multiple sub-matrices can be reshaped and aligned to the tensor-style. However, the problem in (15) 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.

Before introducing our solution, we first investigate the relationship between tensor CP decomposition and frontal slice decomposition. As shown in Fig.8, given a 3-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, $\mathbf{C} \in \mathbb{R}^{K \times R}$

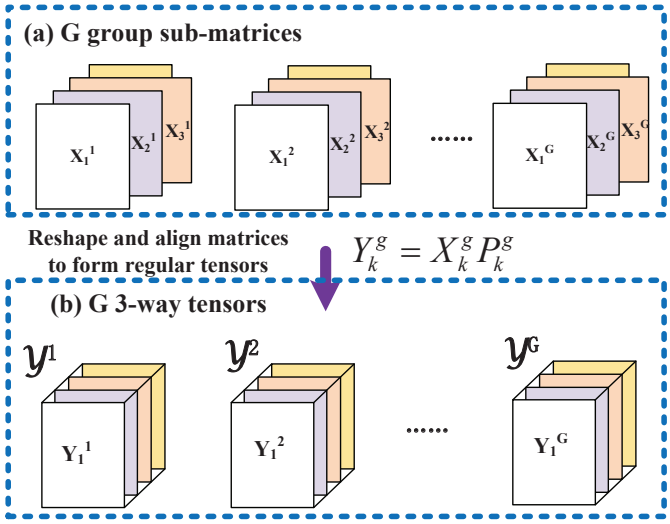


Fig. 7. Transform G groups of sub-matrices to G tensors.

are the factor matrices in CP decomposition in Fig.8(a). The frontal slice can be decomposed as $\mathbf{X}_k = \mathbf{A}\mathbf{\Sigma}_k\mathbf{B}^T$ where $\mathbf{\Sigma}_k = diag\,(\mathbf{C}_{k:})$ and $\mathbf{C}_{k:}$ is the k-th row of the factor matrix $\mathbf{C}$, as shown in Fig.8(b). This relationship provides the way to recover the group of matrices through tensor completion.

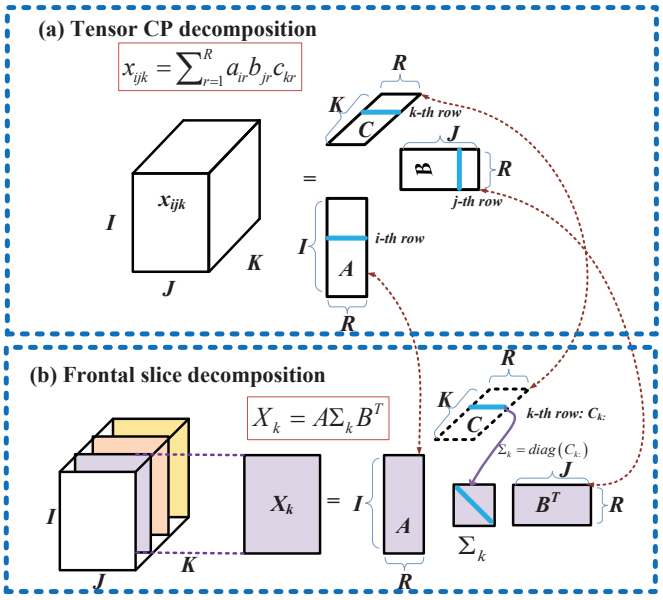


Fig. 8. The relationship between tensor CP decomposition and frontal slice decomposition.

The problem in (15) 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 reflect 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. This translation makes the formulation in (15) satisfy the relationship between tensor CP decomposition and frontal slice decomposition.

Therefore, utilizing the above relationship, the problem can be transformed to the following tensor completion problem:

$$\begin{aligned} &\min_{\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g} f\,(\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g) \\ &\text{s.t. } f\,(\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g) = \tfrac{1}{2}\sum_{g=1}^{G}\left(\left\|(\mathcal{Y}^g - [\![\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g]\!])_{\Omega^g}\right\|_F^2\right) \end{aligned} \tag{16}$$

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, we utilize the solution in [34] to solve the tensor completion to obtain the optimal factor matrices $\mathbf{U}^g$,$\mathbf{V}^g$,$\mathbf{C}^g$ by minimizing the loss function in (16).

After obtaining $\mathbf{U}^g$,$\mathbf{V}^g$,$\mathbf{C}^g$, the reshaped slice can be recovered through following calculation $\hat{\mathbf{Y}}_k^g = \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T$ 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 $\hat{\mathbf{X}}_k^g = \mathbf{Y}_k^g(\mathbf{P}_k^g)^T = \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T(\mathbf{P}_k^g)^T$.

VII. Complete Solution

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 $G-1$ timespots besides the timespots 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 (10) over $\mathbf{P}_k^g$ for fix $\mathbf{U}^g, \mathbf{\Sigma}_k^g, \mathbf{V}^g$. Step 5 builds the tensor with the shaped 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 $\min_{\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g} f\,(\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g) = \frac{1}{2}\left\|(\mathcal{Y}^g - [\![\mathbf{U}^g,\mathbf{V}^g,\mathbf{C}^g]\!])_{\Omega^g}\right\|_F^2$. Step 7 builds the diagonal matrix needed in the matrix decomposition $\mathbf{\Sigma}_k^g \leftarrow dig\,(\mathbf{C}_{k:}^g)$ 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 $c_{ij} = a_{ij}b_{ij}$. $\mathbf{X}_k^g = \mathbf{M}_k^g \odot \mathbf{X}_k^g + (1 - \mathbf{M}_k^g) \odot \mathbf{U}^g\mathbf{\Sigma}_k^g(\mathbf{V}^g)^T(\mathbf{P}_k^g)^T$ guarantees that the sample entry already measured remains unchanged and only the missing data are updated during the iterative procedure.

VIII. Performance Evaluations

We use the public traffic trace data Abilene [35] to evaluate the performance of our proposed Reshape-Align scheme. Two different metrics are considered: Error Ratio and Mean Absolute Error (MAE), which are defined as Table I.

Algorithm 1 Complete reshape and align traffic recovery algorithm

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

TABLE I
PERFORMANCE METRIC

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

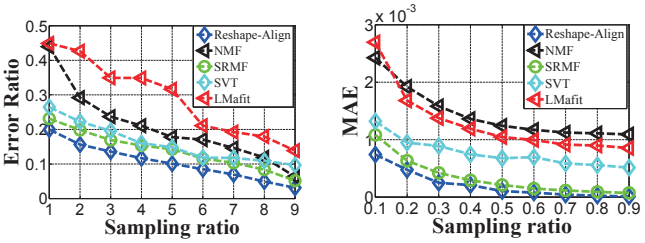
In the table, $x_{k:ij}$ and $\hat{x}_{k:ij}$ denote the raw data and the recovered data at (i,j)-th element of the matrix X_k where $1 \le i \le n, 1 \le j \le m_k$, and $1 \le k \le K$. Only entries not observed $(i,j) \in \bar{\Omega}_k$ are counted in the Error Ratio. Different from Error Ratio,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

Although some limited very recent studies consider the traffic data recovery through tensor completion, they cannot be applied in the practical network with dynamic measurement rate. Therefore, we implement four matrix completion algorithms for the performance comparison:NMF [36], $SRMF$ [11], SVT [37], $LMaFit$ [38].

To align measurement data under different measurement rate 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.

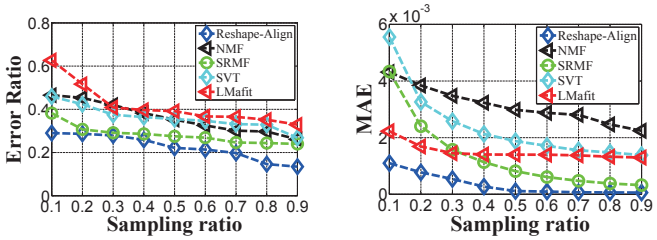
According the time alignment requirement in Section V, different measurement rates will result in different partitions. We take 3 measurement scenarios as examples to show the performance: 1) The measurement rates are different in different days while the measurement rate of the same day is the same. 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.

Fig.9 shows the performance in terms of error ratio and MAE with different sampling ratios. Note that, sampling ratio is the fraction of the total sampling entries to the total



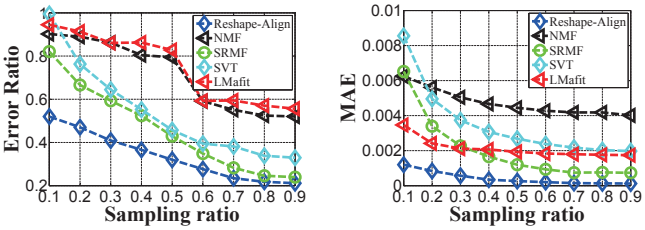
(a) Error Ratio (b) MAE

Fig. 9. Recovery performance under Scenario 1.



(a) Error Ratio (b) MAE

Fig. 10. Recovery performance under Scenario 2.



(a) Error Ratio (b) MAE

Fig. 11. Recovery performance under Scenario 3.

number of matrix entries given a measurement(sampling) rate. As expected, with the increase of the sampling ratio thus sample data, the error ratio and MAE 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, our Reshape-Align scheme achieves the best recovery performance with the lowest error ratio and MAE in all the figures. Among all the matrix completion algorithms, SRMF achieves the best performance. Besides using 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.

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. In Scenario 3, the performance gap between our Reshape-

Align scheme and other matrix completion algorithms becomes large which demonstrates that Reshape-Align scheme can well exploit multi-dimensional correlations hidden in the traffic data to accurately recover the missing data even with a short time period.

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 our scheme achieves significantly better performance compared with the state of art matrix-completion algorithms to handle the missing data.

References

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

[2] A. Gunnar, M. Johansson, and T. Telkamp, "Traffic matrix estimation on a large ip backbone: a comparison on real data," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 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 Performance Evaluation Review*, 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 Computer Communication Review*, vol. 32, no. 4, pp. 161–174, 2002.

[5] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "Opentm: traffic matrix estimator for openflow networks," in *International Conference on Passive and Active Network Measurement*. Springer, 2010, pp. 201–210.

[6] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *ACM SIGMETRICS*, 2003.

[7] Y. Zhang, M. Roughan, C. Lund, and D. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," in *IEEE/ACM Trans. Netw*, 2005, pp. 947–960.

[8] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *Acm Sigcomm Computer Communication Review*, vol. 34, no. 4, pp. 219–230, 2004.

[9] Y.Vardi, "Network tomography," *J. Amer. Statist. Assoc.*, vol. vol. 91, no. 433,, p. pp. 365377, 1996.

[10] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," *ACM IMW*, 2002.

[11] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *Networking IEEE/ACM Transactions on*, vol. 20, no. 3, pp. 662 – 676, 2012.

[12] M. Mardani and G. Giannakis, "Robust network traffic estimation via sparsity and low rank," in *IEEE ICASSP*, 2013.

[13] R. Du, C. Chen, B. Yang, and X. Guan, "Vanet based traffic estimation: A matrix completion approach," in *IEEE GLOBECOM*, 2013.

[14] G. Gürsun and M. Crovella, "On traffic matrix completion in the internet," in *ACM IMC 2012*.

[15] Y.-C. Chen, L. Qiu, Y. Zhang, G. Xue, and Z. Hu, "Robust network compressive sensing," in *ACM MobiCom*, 2014.

[16] K. Xie, X. Ning, X. Wang, D. Xie, J. Cao, G. Xie, and J. Wen, "Recover corrupted data in sensor networks: a matrix completion solution," *IEEE Transactions on Mobile Computing, DOI:10.1109/TMC.2016.2595569*, 2016.

[17] K. Xie, L. Wang, X. Wang, G. Xie, G. Zhang, D. Xie, and J. Wen, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *IEEE INFOCOM*, 2015.

[18] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, and G. Zhang, "Accurate recovery of internet traffic data: A tensor completion approach," in *IEEE INFOCOM*, 2016.

[19] T. Kolda and B. Bader, "The tophits model for higher-order web link analysis," in *Workshop on link analysis, counterterrorism and security*, vol. 7, 2006, pp. 26–29.

[20] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an architecture for never-ending language learning." in *AAAI*, vol. 5, 2010, p. 3.

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

[22] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *Signal Processing Magazine, IEEE*, vol. 32, no. 2, pp. 145–163, 2015.

[23] S. Aja-Fernández, R. de Luis Garcia, D. Tao, and X. Li, *Tensors in image processing and computer vision*. Springer Science & Business Media, 2009.

[24] H. Zhou, D. Zhang, K. Xie, and Y. Chen, "Spatio-temporal tensor completion for imputing missing internet traffic data," in *IEEE IPCCC*, 2015.

[25] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[26] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

[27] E. Acar and T. G. Dunlavy, Daniel M.and Kolda, "A scalable optimization approach for fitting canonical tensor decompositions," *Journal of Chemometrics*, vol. 25, no. 2, p. 6786, 2011.

[28] 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.

[29] J. 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.

[30] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[31] L. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[32] R. A. Harshman, "Parafac2: Mathematical and technical notes," *UCLA working papers in phonetics*, vol. 22, no. 3044, p. 122215, 1972.

[33] B. F. Green, "The orthogonal approximation of an oblique structure in factor analysis," *Psychometrika*, vol. 17, no. 4, pp. 429–440, 1952.

[34] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications." *Siam Review*, vol. 51, no. 3, pp. 455–500, 2009.

[35] "The abilene observatory data collections. http://abilene. internet2.edu /observatory/data-collections.html."

[36] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, 2001, pp. 556–562.

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

[38] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012.