# Expectile Tensor Completion to Recover Skewed Network Monitoring Data

Kun Xie[1,2], Siqi Li[1], Xin Wang[3], Gaogang Xie[4,5], Yudian Ouyang[1]
[1] The College of Computer Science and Electronics Engineering, Hunan University, Changsha, China
[2] The Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China
[3] The Department of Electrical and Computer Engineering, State University of New York at Stony Brook, USA
[4] The Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
[5] The School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing, China

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488919

*Abstract*—**Network applications, such as network state tracking and forecasting, anomaly detection, and failure recovery, require complete network monitoring data. However, the monitoring data are often incomplete due to the use of partial measurements and the unavoidable loss of data during transmissions. Tensor completion has attracted some recent attentions with its capability of exploiting the multi-dimensional data structure for more accurate un-measurement/missing data inference. Although conventional tensor completion algorithms can work well when the application data follow the symmetric normal distribution, it cannot well handle network monitoring data which are highly skewed with heavy tails. To better follow the data distribution for more accurate recovery of the missing entries with large values, we propose a novel expectile tensor completion (ETC) formulation and a simple yet efficient tensor completion algorithm without hard-setting parameters for easy implementation. From both experimental and theoretical ways, we prove the convergence of the proposed algorithm. Extensive experiments on two real-world network monitoring datasets demonstrate the effectiveness of the proposed ETC.**

*Index Terms*—**Tensor completion, Skewed network monitoring data, Asymmetric least squares**

## I. Introduction

It is very important to monitor the network states for the troubleshoot of network incidents and performance issues such as packet losses, latency spikes, and traffic burst [1], [2]. However, network monitoring data are often incomplete due to the use of sampling for low cost network measurement and unavoidable data loss under severe communication and system conditions.

As part of the applications of network state tracking and forecasting [6], anomaly detection [7] and failure recovery are highly sensitive to the missing of data, either because of the un-measurement or the loss. This makes the accurate recovery of missing monitoring data from partially observed samples an important procedure in network monitor systems.

The quality of data recovery highly relies on the inference technique to recover the complete data from partial measurements, for which various studies have been made. Originally, designed based on purely spatial [8], [9] or temporal [10] information, the data recovery performance of most known approaches is low. To utilize both spatial and temporal information, matrix completion [3], [5], [11]–[14] has been exploited to recover the missing data from a low-rank matrix. Although these approaches present good performance under low data missing ratio, when the missing ratio is large, the recovery performance suffers, as the two-dimensional matrix-based data analysis has the limitation in extracting information.

For better data recovery, a few recent studies [4], [26]–[28] start to model the network monitoring data as a higher dimensional array called *tensor*, and propose algorithms based on tensor completion for more accurate missing data recovery. Tensors are the high-order generalization of vectors and matrices. Tensor-based multilinear data analysis [15]–[17] has shown that tensor models can take full advantages of the multilinear structures to provide better data understanding. Despite its effectiveness, we argue that current tensor completion algorithms are hindered by following two critical limitations to handle the network monitoring data.

**(a) Unable to accurately estimate the missed data with large values**. Conventional tensor completion algorithms usually adopt symmetric least squared loss function to minimize the aggregate loss during the estimation of all observed samples. They can work well when the application data follow symmetric normal distributions. However, network monitoring data are highly skewed with heavy tails. Our experimental results on real network monitoring datasets also reveal that the missing of network monitoring data with large values are seriously underestimated using the conventional tensor completion algorithms. However, in network traffic monitoring, large data items correspond to elephant flows, which are especially important in many network applications, including congestion control to dynamically schedule elephant flows [18], network capacity planning [19], anomaly detection [20], and caching of forwarding table entries [21].

**(b) Difficult to set the parameters in tensor completion.** To solve tensor completion problem, Stochastic Gradient Descent (SGD) [24] is generally adopted to pursue the optimal solution. While in the practice implementation, we found the crucial parameter, the *learning rate*, is very hard to determine. A small learning rate takes long time to converge, while a large rate may easily overshoot the optimal solution. Moreover, an inappropriate learning rate will also increase the deviation of the completion result.


978-1-6654-0325-2/21/$31.00 ©2021 IEEE

To well utilize tensor's strong ability of data representing and information extraction, in this paper, we model network monitoring data as a 3-dimensional tensor and design a novel expectile tensor completion to overcome the above limitations. The main technique contributions are listed as follows.

• To accurately estimate the missed network monitoring data whose values are big, we formulate the tensor completion problem by replacing the symmetric least squares loss function in conventional tensor completion problem with a loss function similar to those used in expectile regression. The expectile tensor completion formulation helps to better follow the data distribution for more accurate recovery of the large missing entries.

• To solve the expectile tensor completion problem, we propose a simple yet efficient tensor completion algorithm without hard-setting parameters for easy implementation. From both experimental and theoretical way, we prove the convergence of the proposed algorithm.

• We conduct comprehensive experiments to comparatively evaluate and demonstrate the effectiveness of the proposed method. The experimental results demonstrate that our ETC can achieve significantly better recovery accuracy especially on the large missing entries. When the sampling ratio is low at 10%, the recovery error ratio on the large missing data is around 0.2 using ETC, but is 0.7 using the best reference tensor completion algorithms, which is 3.5 times larger.

The rest of the paper is organized as follows. We introduce the related work and preliminaries in Section II and Section III, respectively. The system model and problem are presented in Section IV. We present our new expectile tensor completion problem and algorithm in Section V and Section VI, respectively. Finally, we evaluate the performance of the proposed expectile tensor completion algorithm through extensive experiments in Section VII, and conclude the work in Section VIII.

## II. Related Work

Recovering un-measurement data with partial measurement samples is a very important procedure for network monitoring. Besides original algorithms using purely spatial or temporal [8]–[10], and matrix-based algorithms [3], [5], [11]–[14] using simple spatio-temporal features, a few recent studies [4], [26]–[28] propose to apply the tensor completion to capture more spatio-temporal features in the network monitoring data for more accurate recovery.

These initial efforts demonstrate the potential of applying tensor-based schemes for better monitoring data recovery. However, we find that these techniques are designed based on symmetric least squares loss function, where solutions are obtained from the estimate of the conditional mean of the observation samples. However, as network monitoring data are highly skewed with heavy tails, these current techniques face a serious problem to underestimate the large missing entries.

Besides inferring the network monitoring data, in terms of methodologies, many variants [29]–[33] of tensor completion method have been developed to capture the global data structure for recovering the missing data via tensor factorization. These variants still adopt symmetric least squares loss function to describe the problem. They may still suffer from insufficiently capturing the data distribution feature of real network monitoring data and underestimating the large missing entries. Moreover, current tensor completion problems are usually solved based on either Alternating Least Squares (ALS) [37] or SGD , which may suffer from the problems of large computation cost or hard parameter setting.

To conquer the limitations in current studies, we propose to formulate a novel expectile tensor completion problem, which can better capture the data distribution of network monitoring data to accurately estimate the large missing entries. Moreover, instead of ALS and SGD, we propose a simple yet efficient tensor completion algorithm to iteratively solve the problem with well designed update rules to ensure the convergence. Experiments demonstrate the effectiveness and efficiency of the proposed techniques.

## 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 $i$th entry of vector $\mathbf{x}$ is denoted by $x_i$, element $(i, j)$ of a matrix $\mathbf{X}$ is denoted by $x_{ij}$, and element $(i, j, k)$ of a third-order tensor $\mathcal{X}$ is denoted by $x_{ijk}$. The row (column) vectors of a matrix are denoted by the symbolic name of the matrix with indexes in subscript. For example, the $i$-th row (column) vector of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_{i:}$ $(\mathbf{a}_{:i})$. The transpose, inverse, and Frobenius norm of a matrix $\mathbf{X}$ are denoted by $\mathbf{X}^T$, $\mathbf{X}^{-1}$ and $\|\mathbf{X}\|_F$. Some preliminaries in this paper can be found in [34]–[36].

**Definition 1 (Khatri-Rao Product).** Given two matrices with the same number of columns $\mathbf{A} \in \mathbb{R}^{\mathrm{m} \times n}$ and $\mathbf{B} \in \mathbb{R}^{p \times n}$, the Khatri-Rao product between these two matrices can be defined as :

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathrm{a}_{11} b_{11} & \mathrm{a}_{12} b_{12} & & \mathrm{a}_{1n} b_{1n} \\ \vdots & \vdots & & \vdots \\ \mathrm{a}_{11} b_{p1} & \mathrm{a}_{12} b_{p2} & & \mathrm{a}_{1n} b_{pn} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} b_{11} & a_{m2} b_{12} & & \mathrm{a}_{1n} b_{pn} \\ \vdots & \vdots & & \vdots \\ a_{m1} b_{p1} & a_{m2} b_{p2} & & a_{mn} b_{pn} \end{bmatrix} \quad (1)$$

**Definition 2 (Tensor).** A tensor, also has the name of $N$th-order or $N$-way tensor, is the higher-order generalization of a vector (one-way tensor) and a matrix (two-way tensor). A $N$-way tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where $N$ is the mode of the tensor and the element in a tensor can be written as $x_{i_1, i_2, \ldots, i_N}$, $i_n \in 1, 2, \ldots, I_n$, $1 \leq n \leq N$.

**Definition 3 (Slice).** The slice of an $N$-order tensor is an $(N-1)$-order tensor with the index of a particular mode fixed.

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.

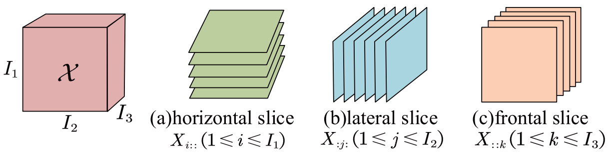



Figure 1. Tensor slide.

**Definition 4 (Unfolding).** Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, a mode-$k$ vector $\mathbf{v}$ is defined as the vector that is obtained by fixing all indexes of $\mathcal{X}$ but varying the mode-$k$ index. We refer to the set of vectors associated different modes of $\mathcal{X}$ as the vector space of mode. The mode-$k$ unfolding, or matricization, of $\mathcal{X}$, denoted by $\mathbf{X}_{(k)}$. Fig. 2 shows an unfolding procedure of a 3-way tensor, which involves the tensor dimensions $I_1$, $I_2$, $I_3$ in a cyclic way.

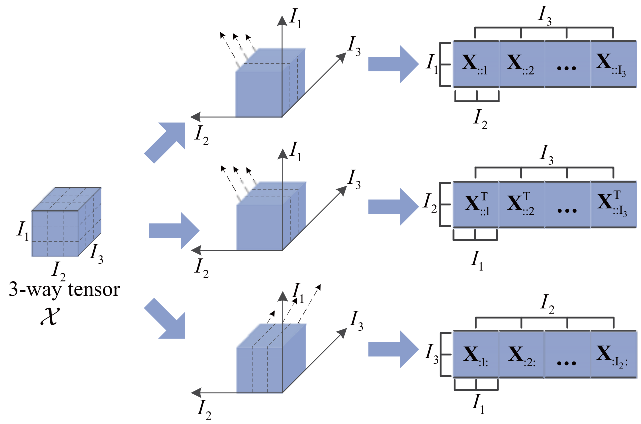



Figure 2. Unfolding of the $(I_1 \times I_2 \times I_3)$ tensor $\mathcal{X}$ to the $(I_1 \times I_2 I_3)$ matrix $\mathbf{X}_{(1)}$, the $(I_2 \times I_3 I_1)$ matrix $\mathbf{X}_{(2)}$, and the $(I_3 \times I_1 I_2)$ matrix $\mathbf{X}_{(3)}$.

The outer product of three column vectors $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ is a rank one tensor given by $(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})_{ijk} = a_i b_j c_k$ for all values of the indexes.

**Definition 5.** *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 $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ can be expressed as*

$$\mathcal{X} = \sum\nolimits_{r=1}^{R} \mathbf{a}_{:r} \circ \mathbf{b}_{:r} \circ \mathbf{c}_{:r}, \tag{2}$$

*with an entry calculated as*

$$x_{ijk} = \sum\nolimits_{r=1}^{R} a_{ir} b_{jr} c_{kr} \tag{3}$$

*where $R$ is the tensor rank with $R > 0$, $a_{ir}$, $b_{jr}$, $c_{kr}$ are the $i$-th, $j$-th, and $k$-th entry of vectors $\mathbf{a}_{:r} \in \mathbb{R}^{I_1}$, $\mathbf{b}_{:r} \in \mathbb{R}^{I_2}$, and $\mathbf{c}_{:r} \in \mathbb{R}^{I_3}$, respectively.*

Fig. 3 illustrates the CP decomposition. By collecting the vectors in the rank one components, we have tensor $\mathcal{X}$'s factor matrices $\mathbf{A} = [\mathbf{a}_{:1}, \cdots, \mathbf{a}_{:R}] \in \mathbb{R}^{I_1 \times R}$, $\mathbf{B} = [\mathbf{b}_{:1}, \cdots, \mathbf{b}_{:R}] \in \mathbb{R}^{I_2 \times R}$, and $\mathbf{C} = [\mathbf{c}_{:1}, \cdots, \mathbf{c}_{:R}] \in \mathbb{R}^{I_3 \times R}$. Using the factor matrices, we can rewrite the CP decomposition as follows.

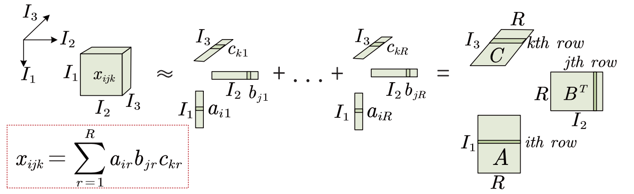



Figure 3. CP decomposition of a 3-way tensor as sum of $R$ outer products (rank one tensors).

$$\mathcal{X} = \sum\nolimits_{r=1}^{R} \mathbf{a}_{:r} \circ \mathbf{b}_{:r} \circ \mathbf{c}_{:r} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{4}$$

An entry $x_{ijk}$ can be calculated as the sum of the dot product of row vectors $\mathbf{a}_{i:}$, $\mathbf{b}_{j:}$, and $\mathbf{c}_{k:}$. That is, $x_{ijk} = \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:}$, where $\mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$ is the dot product of the three row vectors $\mathbf{a}_{i:}$, $\mathbf{b}_{j:}$, and $\mathbf{c}_{k:}$ with $\mathbf{a}_{i:} \in \mathbb{R}^{1 \times R}$, $\mathbf{b}_{j:} \in \mathbb{R}^{1 \times R}$, and $\mathbf{c}_{k:} \in \mathbb{R}^{1 \times R}$.

## IV. System Model and Problem

In this section, we first present the system model and problem, then analyze the drawbacks in conventional tensor completion solutions.

### A. Problem Description

We model the network monitoring data as a 3-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ (Fig. 2), where $I_1$, $I_2$, and $I_3$ correspond to the number of origin nodes, the number of destination nodes, and time intervals monitored, respectively. Each entry $x_{ijk}$ indicates the end-to-end network monitoring data from the origin node $i$ to the destination node $j$ in the time slot $k$.

If there are no network measurement data between a pair of nodes in a given time slot, it leaves the corresponding entry in $\mathcal{X}$ empty. All observed entries are denoted by $\Omega = \{i, j, k | x_{ijk} \ is \ known\}$. To reduce the cost, partial node pairs are often monitored to reduce the measurement load. In addition, there are unavoidable data losses upon severe communication conditions. Therefore, $\mathcal{X}$ is generally a sparse and incomplete tensor. As many network engineering tasks require the complete network performance information, the accurate reconstruction of unmeasured data from partial traffic measurements becomes a key problem, and we refer this problem as the network monitoring data recovery problem.

### B. Conventional Tensor Completion Approach

To recover the tensor $\mathcal{X}$ based on its measurement samples, with the factorization based on the CP decomposition, the network monitoring data recovery problem is defined as

$$\min_{\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}} F(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}) \tag{5}$$

where

$$\begin{aligned} &F(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}) = \tfrac{1}{2} \| (x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:})_{\Omega} \|_F^2 \\ &+ \tfrac{1}{2} \lambda \left( \sum\nolimits_i \|\mathbf{a}_{i:}\|^2 + \sum\nolimits_j \|\mathbf{b}_{j:}\|^2 + \sum\nolimits_k \|\mathbf{c}_{k:}\|^2 \right) \end{aligned} \tag{6}$$

In Eq. (5) and Eq. (6), $\mathbf{a}_{i:}$ is the $i$th row of factor matrix $\mathbf{A}$, $\mathbf{b}_{j:}$ is the $j$th row of factor matrix $\mathbf{B}$, $\mathbf{c}_{k:}$ is the $k$th row of factor matrix $\mathbf{C}$. The problem (5) above requires the finding of the factor matrices $\mathbf{A} \in \mathbb{R}^{I_1 \times R}$, $\mathbf{B} \in \mathbb{R}^{I_2 \times R}$, and $\mathbf{C} \in \mathbb{R}^{I_3 \times R}$ to approximate the tensor $\mathcal{X}$ with the minimum loss $F(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:})$. $\lambda$ is the regularization coefficients, $\lambda\left(\sum_i \|\mathbf{a}_{i:}\|^2 + \sum_j \|\mathbf{b}_{j:}\|^2 + \sum_k \|\mathbf{c}_{k:}\|^2\right)$ is the regularization item to prevent over-fitting problem.

Among various existing optimization methods for the loss minimization, ALS [37] and SGD [24] are two kinds of methods. The computation cost of ALS is much higher than that of SGD, thus tensor completion algorithms usually adopt the SGD method to train the factor matrices. It randomly selects a sample $m_{ijk}$ and applies the update rules (7) to update the factor matrices iteratively.

$$\begin{aligned} \mathbf{a}_{i:} &\leftarrow \mathbf{a}_{i:} - \epsilon N \frac{\partial}{\partial \mathbf{a}_{i}} F_{ijk}(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}) \\ \mathbf{b}_{j:} &\leftarrow \mathbf{b}_{j:} - \epsilon N \frac{\partial}{\partial \mathbf{b}_{j}} F_{ijk}(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}) \\ \mathbf{c}_{k:} &\leftarrow \mathbf{c}_{k:} - \epsilon N \frac{\partial}{\partial \mathbf{c}_{k}} F_{ijk}(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}) \end{aligned} \tag{7}$$

where $F_{ijk}(\mathbf{a}_{i:}, \mathbf{b}_{j:}, \mathbf{c}_{k:}) = \frac{1}{2}(x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:})^2 + \frac{\lambda}{2N_i}\|\mathbf{a}_{i:}\|^2 + \frac{\lambda}{2N_j}\|\mathbf{b}_{j:}\|^2 + \frac{\lambda}{2N_k}\|\mathbf{c}_{k:}\|^2$ is the local loss at sample location $(i, j, k)$ with $N_i$, $N_j$, $N_k$ denoting the number of observed entries on the $i$-th origin node, $j$-th destination node, and $k$-th time interval in the $\mathcal{X}$.

The iteration will continue until the loss difference between two consecutive iterations is smaller than a given threshold value or the given number of iterations is reached. The parameter $\epsilon$ in Eq. (7) is the learning rate.

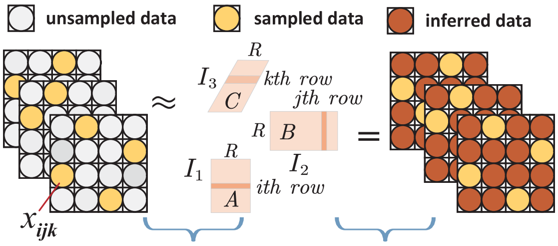



Figure 4. Overview of tensor completion.

Fig. 4 illustrates the two main steps involved to solve the CP based tensor completion problem. 1) **tensor factorization**: training the latent feature vectors using the partial measurement samples through SGD, 2) **missing data recovery**: inferring missing tensor entries using the trained latent feature vectors. After getting three latent feature matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, the network monitoring tensor can be recovered by

$$\hat{\mathcal{X}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \tag{8}$$

with the recovered entry $\hat{x}_{ijk} = \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$.

## C. Drawbacks of The Conventional Tensor Completion

Despite its effectiveness, conventional tensor completion algorithm has following inherent limitations in its model design.

**(a) Unable to accurately estimate the large missing entries in the monitoring tensor**. In the conventional tensor completion problem (in Eq. (5)), the symmetric least squares loss function is applied as a metric to guide the training of the factor matrices, with goal of minimizing the aggregate loss from the estimation of all observed samples $\min_{\mathbf{a}_{i:} \mathbf{b}_{j:} \mathbf{c}_{k:}} \sum_{i,j,k \in \Omega} (x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:})^2$. It provides the solutions that estimate the conditional mean of the observation samples. Therefore, although conventional tensor completion algorithms can work well when the application data follow the symmetric normal distribution, it can not work well to handle network monitoring data as they are highly skewed with heavy tails. Following we present the experimental results to demonstrate the limitations.

**Real data distribution.** We study the data distribution of two network traffic monitoring datasets: Abilene [42] and GÈANT [43]. We found that 90% measured data are smaller than 0.0076, yet the largest measured data value is 0.8857 in Abilene, and 90% measured data are smaller than $2.8343e4$, while the largest measured value is $9.2181e9$ in GÈANT. The network data are highly skewed with heavy tails.

**Estimation performance under conventional tensor completion.** In the tensor completion problem formulated in Eq. (5), measurement samples are applied to train the factor matrices, which are used to recover the monitoring tensor through Eq. (8). Following the update rules in Eq. (7), we implement the SGD based tensor completion algorithm on the traffic datasets with the sampling ratio set to be 30% and present the recovery result in Fig. 5.

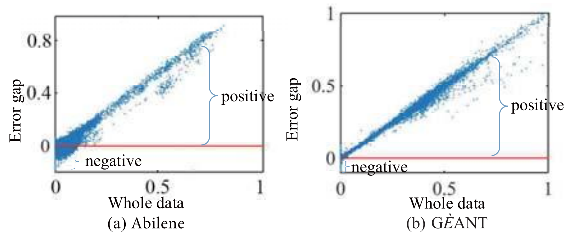



Figure 5. Error gap between origin data and recovery data. The values are normalized to [0, 1] for the presentation convenience.

In these figures, the horizontal axis displays the individual data values of the monitoring data, and the vertical axis records the error gap (also called residual in next following section) between a real value $x_{ijk}$ and its estimate $\mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:}$, that is, $x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:}$. In Fig. 5, we can find that the error gap increases with the increase of data value. For the data with small values, the error gaps are negative with small absolute values, which indicates that the tensor completion algorithm over-estimates the data. While for those data with large values, the error gap is positive and increases with the increase of data values. That is, the tensor completion algorithm underestimates the data.

**(b) Difficulty in setting the parameters for the tensor completion.** Learning rate $\varepsilon$ is an important parameter for effectively training factor matrices using the measurement samples. It determines the size of the steps which the SGD optimizer takes along the direction downhill until reaching a (local) minimum (valley). If the learning rate is low, the training process is more reliable, but it will take a long period of time to converge. If the learning rate is high, the training process may not converge or even diverge. The update of the factor matrices will be so big that the tensor factorization overshoots the optimal solution and makes the loss worse. Some studies also indicate that an inappropriate learning rate will also increase the deviation of the completion result.

## V. Expectile Tensor Completion problem

Experimental results in Fig. 5 show that the error gaps (residuals) are negative on small monitoring data, but are positive on large monitoring data. Following these observations, we propose to formulate a new expectile tensor completion problem to well capture the data distribution.

Inspired by the quantile regression [38], [39], Newey and Powell apply different weights to the positive and negative residuals in the least-squares algorithm and propose the asymmetric least squares, named expectile regression [40]. With respect to each $y_i$ and a given $w \in [0,1]$, the $w$th expectile regression is expressed as follows:

$$\min_{y_i} \sum_{i=1}^{n} |w - I(y_i \le 0)|\,(y_i)^2 \tag{9}$$

where $I$ is the indicator function and $I$ equals to 1 when $y_i \le 0$ otherwise $I$ equals to 0. And (9) can be further expressed as:

$$\min_{y_i} \sum_{i=1}^{n} (1-w)\,(y_i)_-^2 + w\,(y_i)_+^2 \tag{10}$$

where $(y_i)_-$ and $(y_i)_+$ indicate the cases $y_i \le 0$ and $y_i > 0$, respectively. Let

$$\rho(y_i) = |w - I(y_i \le 0)|\,(y_i)^2 \tag{11}$$

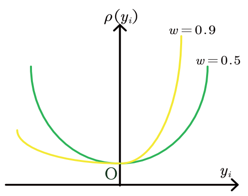



Figure 6. The shape of $\rho(y_i)$.

Fig. 6 illustrates the shape of $\rho(y_i)$. We can see that when $w > 0.5$, the weight in the positive residual is larger than the weight in the negative residual, and vice versa when $w \le 0.5$. Obviously, when we set $w = 0.5$, the asymmetric least squares equals to the normal symmetric least squares.

Based on expectile regression, we formulate the expectile tensor completion problem as follows:

$$\min_{\mathbf{a}_{i:},\mathbf{b}_{j:},\mathbf{c}_{k:}} F(\mathbf{a}_{i:},\mathbf{b}_{j:},\mathbf{c}_{k:}) \tag{12}$$

where

$$\begin{aligned} F(\mathbf{a}_{i:},\mathbf{b}_{j:},\mathbf{c}_{k:}) &= \tfrac{1}{2}\textstyle\sum_{(i,j,k)\in\Omega} \rho(x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:}) \\ &+ \tfrac{1}{2}\lambda\left(\textstyle\sum_i \|\mathbf{a}_{i:}\|^2 + \sum_j \|\mathbf{b}_{j:}\|^2 + \sum_k \|\mathbf{c}_{k:}\|^2\right) \end{aligned} \tag{13}$$

and (13) can be further expressed as:

$$\begin{aligned} F(\mathbf{a}_{i:},\mathbf{b}_{j:},\mathbf{c}_{k:}) &= \tfrac{1}{2}\textstyle\sum_{(i,j,k)\in\Omega} (1-w)\left[(x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:})_-\right]^2 \\ &+ w\left[(x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:})_+\right]^2 \\ &+ \tfrac{\lambda}{2}\left(\textstyle\sum_i \|\mathbf{a}_{i:}\|^2 + \sum_j \|\mathbf{b}_{j:}\|^2 + \sum_k \|\mathbf{c}_{k:}\|^2\right) \end{aligned} \tag{14}$$

In the formulation, we weight each residual $x_{ijk} - \mathbf{a}_{i:} \bullet \mathbf{b}_{j:} \bullet \mathbf{c}_{k:}$ by $w$ or $1-w$, conditioned on whether it is positive or negative.

Our expectile tensor completion in (12) provides a flexible problem formulation for different focuses of the data recovery by setting different $w$. Experimental results in Fig. 5 show that the error gaps (residuals) are negative on small monitoring data, but are positive on large monitoring data. According to the results, if we care more about the large data (i.e., elephant flow), we set $w > 0.5$. If we care more about the small data, we set $w < 0.5$. When we set $w = 0.5$, it equals to the traditional symmetric least squares, which is usually adopted by current tensor completion algorithms. In Section VII, we will show how $w$ impacts the performance of missing data recovery.

## VI. Tensor completion algorithm

Let look at the expectile tensor completion problem in Eq. (12). According to the residual between the real and estimated values, the weight to each sample entries has a different value. We use the weight tensor $\mathcal{P}$ to denote the sample weight, the expectile tensor completion problem with respect to $\mathbf{a}_{i:}$, which is the $i$-th row in factor matrix $\mathbf{A}$, can be further written as:

$$F(\mathbf{a}_{i:}) = \tfrac{1}{2}\Sigma_h \left(\mathbf{P}_{(1)}\right)_{ih} \left(\left(\mathbf{X}_{(1)}\right)_{ih} - \mathbf{a}_{i:}\mathbf{m}_{:h}\right)^2 + \tfrac{\lambda}{2}\|\mathbf{a}_{i:}\|^2 \tag{15}$$

where $\mathbf{P}_{(1)}, \mathbf{X}_{(1)} \in \mathbb{R}^{I_1 \times Z_1}\,(Z_1 = I_3 \times I_2)$ are the mode-1 unfolding matrices of the weight tensor $\mathcal{P}$ and the monitoring tensor $\mathcal{X}$. $\mathbf{m}_{:h}\,(1 \le h \le I_3 \times I_2)$ is the $h$th column in $\mathbf{M}_1$, and $\mathbf{M}_1$ can be further expressed as: $\mathbf{M}_1 = (\mathbf{C} \odot \mathbf{B})^T$ where $\odot$ is Khatri-Rao Product.

Through ALS and SGD, we can design algorithm to solve the problem in Eq. (15). However, as discussed in Section IV, ALS has a large computation cost, while it is hard to set a proper learning rate of SGD. To conquer the limitations, instead of ALS and SGD, we propose a novel expectile tensor completion algorithm which is iteratively executed with new update rules, then present the proof of the convergence of our algorithm.

### A. Algorithm Design

Our tensor completion algorithm executes iteratively until it converges. For the first iteration, we give the update rules as follows:

$$\begin{aligned} \mathbf{a}_{i:}^t &= \mathbf{a}_{i:}^0 * \frac{\left(\mathbf{X}_{(1)}\right)_{i:}\left(\mathbf{M}_1^0\right)^T}{\mathbf{a}_{i:}^0\left(\mathbf{M}_1^0\left(\mathbf{M}_1^0\right)^T + \lambda\mathbf{I}\right)} && t=1 \\ \mathbf{b}_{j:}^t &= \mathbf{b}_{j:}^0 * \frac{\left(\mathbf{X}_{(2)}\right)_{j:}\left(\mathbf{M}_2^0\right)^T}{\mathbf{b}_{j:}^0\left(\mathbf{M}_2^0\left(\mathbf{M}_2^0\right)^T + \lambda\mathbf{I}\right)} && t=1 \\ \mathbf{c}_{k:}^t &= \mathbf{c}_{k:}^0 * \frac{\left(\mathbf{X}_{(3)}\right)_{k:}\left(\mathbf{M}_3^0\right)^T}{\mathbf{c}_{k:}^0\left(\mathbf{M}_3^0\left(\mathbf{M}_3^0\right)^T + \lambda\mathbf{I}\right)} && t=1 \end{aligned} \tag{16}$$

where $\mathbf{a}_{i:}^0$, $\mathbf{b}_{j:}^0$, and $\mathbf{c}_{k:}^0$ are the $i$th row, $j$th row, and $k$th row of initial factor matrices $\mathbf{A}^0$, $\mathbf{B}^0$, and $\mathbf{C}^0$ with values randomly set. In (16), $\mathbf{M}_1^0 \in \mathbb{R}^{R \times Z_1}\,(Z_1 = I_3 \times I_2)$,

$\mathbf{M}_2^0 \in \mathbb{R}^{R\times Z_2}\,(Z_2 = I_3 \times I_1)$, $\mathbf{M}_3^0 \in \mathbb{R}^{R\times Z_3}\,(Z_3 = I_2 \times I_1)$ are expressed as follows:

$$\begin{aligned} \mathbf{M}_1^0 &= \left(\mathbf{C}^0 \odot \mathbf{B}^0\right)^T \\ \mathbf{M}_2^0 &= \left(\mathbf{C}^0 \odot \mathbf{A}^0\right)^T \\ \mathbf{M}_3^0 &= \left(\mathbf{B}^0 \odot \mathbf{A}^0\right)^T \end{aligned} \tag{17}$$

where $\odot$ is Khatri-Rao Product. After the first round, the update rule can be written as follows:

$$\begin{aligned} \mathbf{a}_{i:}^t &= \mathbf{a}_{i:}^{t-1} * \frac{\left(\left(\mathbf{X}_{(1)}\right)_{i:} * \left(\mathbf{P}_{(1)}\right)_{i:}^{t-1}\right)\left(\mathbf{M}_1^{t-1}\right)^T}{\mathbf{a}_{i:}^{t-1}\mathbf{Q}_1^{t-1}} \quad t>1 \\ \mathbf{b}_{j:}^t &= \mathbf{b}_{j:}^{t-1} * \frac{\left(\left(\mathbf{X}_{(2)}\right)_{j:} * \left(\mathbf{P}_{(2)}\right)_{j:}^{t-1}\right)\left(\mathbf{M}_2^{t-1}\right)^T}{\mathbf{b}_{j:}^{t-1}\mathbf{Q}_2^{t-1}} \quad t>1 \\ \mathbf{c}_{k:}^t &= \mathbf{c}_{k:}^{t-1} * \frac{\left(\left(\mathbf{X}_{(3)}\right)_{k:} * \left(\mathbf{P}_{(3)}\right)_{k:}^{t-1}\right)\left(\mathbf{M}_3^{t-1}\right)^T}{\mathbf{c}_{k:}^{t-1}\mathbf{Q}_3^{t-1}} \quad t>1 \end{aligned} \tag{18}$$

In the $t$-th iteration step, the weight tensor is denoted by $\mathcal{P}^{t-1}$. In (18), $\left(\mathbf{P}_{(n)}\right)^{t-1}$ $(n = 1, 2, 3)$ is the mode-$n$ unfolding matrix of the weight tensor $\mathcal{P}^{t-1}$ with its entry $p_{ijk}^{t-1} = w$ if the residual $x_{ijk} - \hat{x}_{ijk}^{t-1}$ is positive value, $p_{ijk}^{t-1} = 1-w$ if the residual $x_{ijk} - \hat{x}_{ijk}^{t-1}$ is negative value. $\hat{x}_{ijk}^{t-1}$ is the estimated entry in the previous iteration step with $\hat{x}_{ijk}^{t-1} = \mathbf{a}_{i:}^{t-1} \bullet \mathbf{b}_{j:}^{t-1} \bullet \mathbf{c}_{k:}^{t-1}$ where $\mathbf{a}_{i:}^{t-1}$, $\mathbf{b}_{j:}^{t-1}$ and $\mathbf{c}_{k:}^{t-1}$ are the $i$th, $j$th, and $k$th row of the factor matrix $\mathbf{A}^{t-1}$, $\mathbf{B}^{t-1}$, $\mathbf{C}^{t-1}$. $\left(\mathbf{P}_{(n)}\right)_{i:}^{t-1}$ is the $i$th row of $\left(\mathbf{P}_{(n)}\right)^{t-1}$. Similarly, $\mathbf{X}_{(1)}$, $\mathbf{X}_{(2)}$, and $\mathbf{X}_{(3)}$ are the mode-1, mode-2, and mode-3 unfolding matrices of the monitoring tensor $\mathcal{X}$. * in (18) represents a scalar product of two row vectors. $\mathbf{Q}_1^{t-1}$, $\mathbf{Q}_2^{t-1}$, $\mathbf{Q}_3^{t-1}$are square matrices, and we have

$$\begin{aligned} \mathbf{Q}_1^{t-1} &= \left(\mathbf{M}_1^{'}\right)^{t-1}\left(\mathbf{M}_1^{t-1}\right)^T + \lambda\mathbf{I} \\ \mathbf{Q}_2^{t-1} &= \left(\mathbf{M}_2^{'}\right)^{t-1}\left(\mathbf{M}_2^{t-1}\right)^T + \lambda\mathbf{I} \\ \mathbf{Q}_3^{t-1} &= \left(\mathbf{M}_3^{'}\right)^{t-1}\left(\mathbf{M}_3^{t-1}\right)^T + \lambda\mathbf{I} \end{aligned} \tag{19}$$

where $\mathbf{I}$ is an identity matrix, $\left(\mathbf{M}_1^{'}\right)^{t-1} \in \mathbb{R}^{R\times Z_1}(Z_1 = I_3 \times I_2)$, $\left(\mathbf{M}_2^{'}\right)^{t-1} \in \mathbb{R}^{R\times Z_2}\,(Z_2 = I_3 \times I_1)$, $\left(\mathbf{M}_3^{'}\right)^{t-1} \in \mathbb{R}^{R\times Z_3}\,(Z_3 = I_2 \times I_1)$.

For $(n = 1, 2, 3)$, the $h$-th column of $\left(\mathbf{M}_n^{'}\right)^{t-1}$ can be calculated from following equation.

$$\left(\mathbf{M}_n^{'}\right)_{:h}^{t-1} = \begin{bmatrix} \left(\mathbf{M}_n\right)_{1h}^{t-1} \times \left(\mathbf{P}_{(n)}\right)_{ih}^{t-1} \\ \left(\mathbf{M}_n\right)_{2h}^{t-1} \times \left(\mathbf{P}_{(n)}\right)_{ih}^{t-1} \\ \vdots \\ \left(\mathbf{M}_n\right)_{Rh}^{t-1} \times \left(\mathbf{P}_{(n)}\right)_{ih}^{t-1} \end{bmatrix} \tag{20}$$

Obviously, the update in Eq. (18) does not depend on the learning rate in SGD, we only use the previous iteration result $\mathbf{a}_{i:}^{t-1}$, $\mathbf{b}_{j:}^{t-1}$, $\mathbf{c}_{k:}^{t-1}$ to calculate the current updated $\mathbf{a}_{i:}^t$, $\mathbf{b}_{j:}^t$, $\mathbf{c}_{k:}^t$. As a result, unlike SGD, our algorithm can be easily implemented without requiring any prior knowledge to set the parameters. In the following subsection, we will prove the convergence of the tensor completion algorithm based on the update rules in Eq. (18).

## *B. Convergence Analysis*

We show that the updating rules monotonically reduce the values in the objective function of Eq. (15) .

We update $\mathbf{A}$ with $\mathbf{B}$ and $\mathbf{C}$ fixed. Since the objective function in Eq. (15) is summed over $\mathbf{a}_{i:}$, and in order to simplify the derivation process, we use $\widetilde{\mathbf{X}}$ to replace $\mathbf{X}_{(1)}$, $\widetilde{\mathbf{P}}$ to replace $\mathbf{P}_{(1)}$, and $\widetilde{x}_{ih}$, $\widetilde{p}_{ih}$ represent the $(i, h)$ entry of $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{P}}$ respectively. Thus, the objective function in terms of $\mathbf{a}_i$ can be rewritten as follows:

$$F\left(\mathbf{a}_{i:}\right) = \tfrac{1}{2}\Sigma_h \widetilde{p}_{ih}\left(\widetilde{x}_{ih} - \mathbf{a}_{i:}\mathbf{m}_{:h}\right)^2 + \tfrac{1}{2}\lambda\|\mathbf{a}_{i:}\|^2 \tag{21}$$

where $\mathbf{a}_{i:}$ is the $i$th row of factor matrix $\mathbf{A}$, $\mathbf{m}_{:h}$ is the $h$th column of matrix $\mathbf{M}_1 = \left(\mathbf{C} \odot \mathbf{B}\right)^T$.

Based on Eq. (21), we build an auxiliary function $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right)$ which is expressed as follows:

$$\begin{aligned} G\left(\mathbf{a}_{i:},\ \mathbf{a}_{i:}^t\right) &= F\left(\mathbf{a}_{i:}^t\right) + \nabla F\left(\mathbf{a}_{i:}^t\right)\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)^T \\ &+ \frac{1}{2}\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right) K\left(\mathbf{a}_{i:}^t\right)\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)^T \end{aligned} \tag{22}$$

where $K\left(\mathbf{a}_{i:}^t\right)$ is the diagonal matrix,

$$K_{ih}\left(\mathbf{a}_{i:}^t\right) = \delta_{ih}\left(\mathbf{a}_{i:}^t\mathbf{Q}_1^t\right)_h / a_{ih}^t \tag{23}$$

**Theorem 1.** *Auxiliary function $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right)$ satisfies:*

$$G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right) \geq F\left(\mathbf{a}_{i:}\right), \quad G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}\right) = F\left(\mathbf{a}_{i:}\right) \tag{24}$$

*Proof.* the second-order Taylor expansion to Eq. (21) can be calculated as follows:

$$\begin{aligned} F\left(\mathbf{a}_{i:}\right) &= F\left(\mathbf{a}_{i:}^t\right) + \nabla F\left(\mathbf{a}_{i:}^t\right)\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)^T \\ &+ \frac{1}{2}\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)\left(\mathbf{Q}_1^t\right)\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)^T \end{aligned} \tag{25}$$

Since $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right) = F(\mathbf{a}_{i:})$ is obvious, we only need to prove $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right) \geq F\left(\mathbf{a}_{i:}\right)$, which requires us to prove the following

$$G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right) - F(\mathbf{a}_{i:}) = \left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)\left(K\left(\mathbf{a}_{i:}^t\right) - \mathbf{Q}_1^t\right)\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)^T \geq 0 \tag{26}$$

To prove (26), we construct a matrix $\mathbf{S}^t$ with its $(l, n)\,(1 \leq l, n \leq R)$ entry expressed as follows:

$$\mathbf{S}_{ln}^t\left(\mathbf{a}_{i:}^t\right) = a_{il}^t\left(K\left(\mathbf{a}_{i:}^t\right) - \mathbf{Q}_1^t\right)_{ln} a_{in}^t \tag{27}$$

In following contents, we first prove that $\mathbf{S}^t$ is positive semidefinite, based on which, we can prove $K\left(\mathbf{a}_{i:}^t\right) - \mathbf{Q}_1^t$ is semi-definiteness.

$$\begin{aligned} &\widetilde{\mathbf{x}}_{i:}\mathbf{S}^t\left(\widetilde{\mathbf{x}}_{i:}\right)^T = \Sigma_{ln}\widetilde{x}_{il}s_{ln}^t\widetilde{x}_{in} \\ &= \Sigma_{ln}\widetilde{x}_{il}a_{il}^t\left(K\left(\mathbf{a}_{i:}^t\right)\right)_{ln} a_{in}^t\widetilde{x}_{in} - \widetilde{x}_{il}a_{il}^t q_{ln}^t a_{in}^t\widetilde{x}_{in} \\ &= \Sigma_{ln}a_{il}^t q_{ln}^t a_{in}^t\widetilde{x}_{il}^2 - \widetilde{x}_{il}a_{il}^t q_{ln}^t a_{in}^t\widetilde{x}_{in} \\ &= \Sigma_{ln}q_{ln}^t a_{il}^t a_{in}^t\left(\widetilde{x}_{il}^2 - \widetilde{x}_{il}\widetilde{x}_{in}\right) \\ &= \Sigma_{ln}q_{ln}^t a_{il}^t a_{in}^t\left(\tfrac{1}{2}\widetilde{x}_{il}^2 + \tfrac{1}{2}\widetilde{x}_{in}^2 - \widetilde{x}_{il}\widetilde{x}_{in}\right) \\ &= \tfrac{1}{2}\Sigma_{ln}q_{ln}^t a_{il}^t a_{in}^t\left(\widetilde{x}_{il} - \widetilde{x}_{in}\right)^2 \geq 0 \end{aligned} \tag{28}$$

where $\widetilde{\mathbf{x}}_{i:}$ is the $i$th row of $\widetilde{\mathbf{X}}$, $s_{ln}^t$ is the $(l, n)$ entry of $\mathbf{S}^t$. We can conclude that $\mathbf{S}^t$ is positive semidefinite. As the necessary and sufficient condition for $K\left(\mathbf{a}_{i:}^t\right) - \mathbf{Q}_1^t$ to be positive semidefinite is that $\mathbf{S}^t$ is positive semidefinite, we have $K\left(\mathbf{a}_{i:}^t\right) - \mathbf{Q}_1^t$ is positive semidefinite. Therefore, we have

$$\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)\left(K\left(\mathbf{a}_{i:}^t\right) - \mathbf{Q}_1^t\right)\left(\mathbf{a}_{i:} - \mathbf{a}_{i:}^t\right)^T \geq 0 \tag{29}$$

□

**Theorem 2.** *Following the update rule,*

$$\mathbf{a}_{i:}^{t+1} = \mathbf{a}_{i:}^{t} * \frac{\left(\left(\mathbf{X}_{(1)}\right)_{i:} * \left(\mathbf{P}_{(1)}\right)_{i:}^{t}\right)\left(\mathbf{M}_1^t\right)^T}{\mathbf{a}_{i:}^t \mathbf{Q}_1^t} \tag{30}$$

$F\left(\mathbf{a}_{i:}\right)$ *is non-increasing and converges to its minimum.*

*Proof.* The proof includes two parts. In part 1, we prove that the update rule of $\mathbf{a}_{i:}$ is the solution of $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right)$. In part 2, we prove that $F\left(\mathbf{a}_{i:}\right)$ converges to its minimum.

**Part 1.** We take the derivative $G\left(\mathbf{a}_{i:},\ \mathbf{a}_{i:}^t\right)$ of $\nabla G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right)$ and set it to 0:

$$\mathbf{a}_{i:}^{t+1} = \mathbf{a}_{i:}^{t} - k\left(\mathbf{a}_{i:}^t\right)^{-1} \nabla F\left(\mathbf{a}_{i:}^t\right) \tag{31}$$

where $\nabla F\left(\mathbf{a}_{i:}^t\right)$ is the derivative $F(\mathbf{a}_{i:}^t)$, according to (15), $\nabla F\left(\mathbf{a}_{i:}^t\right)$ can be written as:

$$\begin{aligned}&\nabla F\left(\mathbf{a}_{i:}^t\right) = \\ &\mathbf{a}_{i:}^t\left(\mathbf{M}_1^{'}\right)^t\left(\mathbf{M}_1^t\right)^T - \left(\left(\mathbf{X}_{(1)}\right)_{i:} * \left(\mathbf{P}_{(1)}\right)_{i:}^{t}\right)\left(\mathbf{M}_1^t\right)^T + \lambda \mathbf{a}_{i:}^t\end{aligned} \tag{32}$$

replacing $\nabla F\left(\mathbf{a}_{i:}^t\right)$ in Eq. (31) by Eq. (32), we obtain:

$$\mathbf{a}_{i:}^{t+1} = \mathbf{a}_{i:}^{t} * \frac{\left(\left(\mathbf{X}_{(1)}\right)_{i:} * \left(\mathbf{P}_{(1)}\right)_{i:}^{t}\right)\left(\mathbf{M}_1^t\right)^T}{\mathbf{a}_{i:}^t \mathbf{Q}_1^t} \tag{33}$$

**Part 2.** According to theorem 1, we have $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right) \geq F\left(\mathbf{a}_{i:}\right)$, therefore $F$ is nonincreasing under the update rule:

$$\mathbf{a}_{i:}^{t+1} = arg\ \min G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right) \tag{34}$$

Therefore, we have $F\left(\mathbf{a}_{i:}^{t+1}\right) \leq G\left(\mathbf{a}_{i:}^{t+1}, \mathbf{a}_{i:}^t\right) \leq G\left(\mathbf{a}_{i:}^t, \mathbf{a}_{i:}^t\right) = F(\mathbf{a}_{i:}^t)$. We want to point out that $F\left(\mathbf{a}_{i:}^{t+1}\right) = F\left(\mathbf{a}_{i:}^t\right)$ is satisfied only if $\mathbf{a}_{i:}^t$ is a local minimum of $G\left(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t\right)$. Thus, by iterating the update rules in Eq.(33), we obtain a sequence of estimates that converge to a local minimum $\left(\mathbf{a}_{min}\right)_{i:} = arg\ min_{\mathbf{a}_{i:}} F\left(\mathbf{a}_{i:}\right)$ of the objective function:

$$\begin{aligned}&F\left(\left(\mathbf{a}_{min}\right)_{i:}\right) \leq ...F\left(\mathbf{a}_{i:}^{t+1}\right) \leq F\left(\mathbf{a}^t\right)_{i:} \\ &... \leq F\left(\mathbf{a}_2\right)_{i:} \leq F\left(\mathbf{a}_1\right)_{i:} \leq F\left(\mathbf{a}_0\right)_{i:}\end{aligned} \tag{35}$$

□

We use Fig. 7 to further illustrate the update process under our update rules. Obviously, with the update of $\mathbf{a}_{i:}^t$ calculated in Eq(30), the minimum of the $G(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t)$ approaches the minimum of $F(\mathbf{a}_{i:})$ step by step.

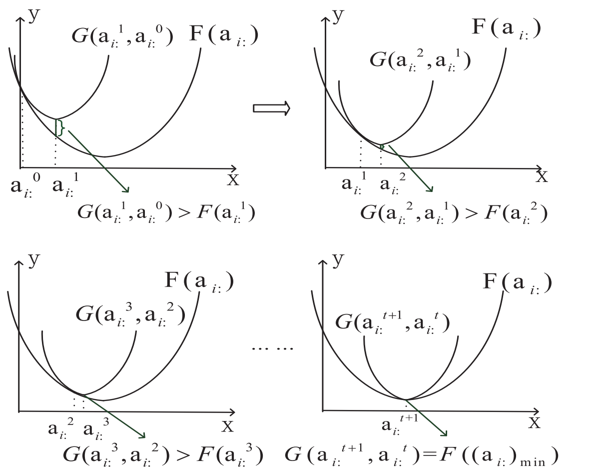



Figure 7. With the update of $\mathbf{a}_{i:}^t$, the minimum of the $G(\mathbf{a}_{i:}, \mathbf{a}_{i:}^t)$ approaches to the minimum of $F(\mathbf{a}_{i:})$ step by step.

## VII. Performance Evaluations

To evaluate the performance of our proposed Expectile Tensor Completion (ETC) model, we use the real network monitoring traces from two sources: the U. S. Internet2 Network (Abilene) and the pan-European research backbone network (GÈANT). Abilene network contains 12 nodes, GÈANT contains 23 nodes. Abilene records the data in every 5 minutes for 168 days. GÈANT records the monitoring data every 15 minutes for 112 days. We set the default sampling ratio to 30%. Due to space constraints and the results of the two datasets are similar, we only present the experimental results on GÈANT.

**Metrics.** We use three metrics to evaluate ETC as shown in Table.I. $x_{ijk}$ and $\hat{x}_{ijk}$ represent the origin data and recovery data at $(i,j,k)$-th element of the monitoring tensor $\mathcal{X}$ respectively, where $1 \leq i \leq I_1$, $1 \leq j \leq I_2$, $1 \leq k \leq I_3$. $\Omega$ denotes the sampled entries in $\mathcal{X}$, while $\bar{\Omega}$ represents the unsampled entries (also named test entries). SER counts the error ratio on sampled entries and TER counts the error ratio on unsampled entries.

Table I
PERFORMANCE METRIC

| | |
|---|---|
| Sampling Error Ratio (SER) | $\frac{\sqrt{\sum_{i,j,k\in\Omega}(x_{ijk}-\hat{x}_{ijk})^2}}{\sqrt{\sum_{i,j,k\in\Omega}x_{ijk}^2}}$ |
| Testing Error Ratio (TER) | $\frac{\sqrt{\sum_{i,j,k\in\bar{\Omega}}(x_{ijk}-\hat{x}_{ijk})^2}}{\sqrt{\sum_{i,j,k\in\bar{\Omega}}x_{ijk}^2}}$ |
| Relative Error (RE) | $\frac{\lvert x_{ijk}-\hat{x}_{ijk}\rvert}{x_{ijk}}\ \left((i,j,k)\in\bar{\Omega}\right)$ |

### *A. Impact of rank R*

To identify the proper rank setting for our monitoring tensor, we vary the rank $R$ and show Loss $\left(\sum_{i,j,k}\left(x_{i,j,k} - \hat{x}_{i,j,k}\right)^2\right)$ in Fig. 8. As expected, an underestimated rank $R$ results in big loss, as CP decomposition with too small $R$ cannot well capture the full structure of the data, while large $R$ brings more computation complexity and thus large computation time. According to Fig. 8, we set $R = 35$ (GÈANT) in our rest experiments.

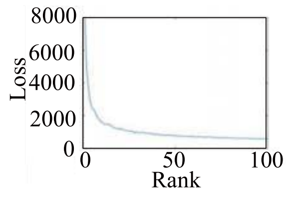



Figure 8. Impact of rank $R$.

### *B. Convergence Behavior*

In Section V, we have presented our tensor completion algorithm and also proven that tensor completion algorithm converges. Fig. 9 further show the convergence behavior of our ETC algorithm. To evaluate the convergence behavior, we use recovery loss, Loss on Sampled Data $\sum_{i,j,k\in\Omega}(x_{i,j,k}-\hat{x}_{i,j,k})^2$ and Inferred Data $\sum_{i,j,k\in\bar{\Omega}}(x_{i,j,k}-\hat{x}_{i,j,k})^2$, where $1 \le i \le I_1$, $1 \le j \le I_2$, $1 \le k \le I_3$, and $x_{i,j,k}$ and $\hat{x}_{i,j,k}$ are the raw and recovered data values. Obviously, in all experiment scenarios with the use of different traces, losses in the ETC implementations with various parameters $(w = 0.1, w = 0.25, w = 0.5, w = 0.75, w = 0.9)$ decrease with iterations and converge to a stable value quickly, which is consistent with Theorem 2 in Section VI-B. This demonstrates that our ETC algorithm in Section VI-A is efficient and effective to solve the expectile tensor completion problem in Eq.(12).

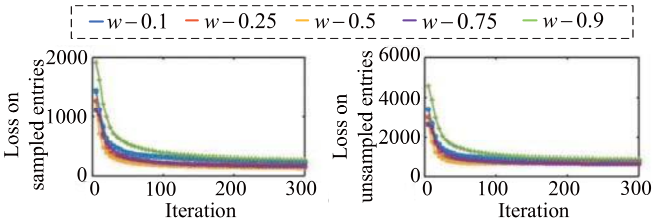



Figure 9. Convergence behavior.

### C. Impact of Parameter $w$

As we discuss in Section V, expectile tensor completion provides a flexible problem formulation for different focuses of the data recovery by setting different $w$. Among the whole dataset, we select two parts to investigate how $w$ impacts the recovery performance. The first is the sub-dataset, denoted by BIG value, consisting of the data entries with the value larger than 0.9 quantile of the whole dataset. The second is the sub-dataset, denoted by SMALL value, consisting of the data entries with the value smaller than 0.1 quantile of the whole dataset.

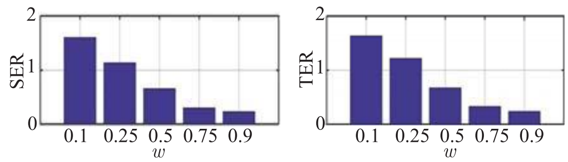



Figure 10. Impact of $w$ on the data with BIG value.

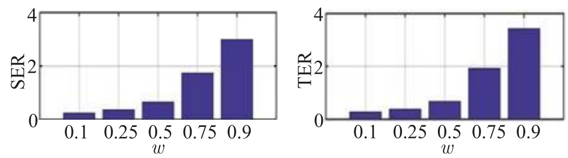



Figure 11. Impact of $w$ on the data with SMALL value.

Fig.10 shows the recovery performance under different $w$ (i.e., 0.1, 0.25, 0.5, 0.75, and 0.9) on the data with BIG values. While Fig.11 shows the recovery performance on the data with small values. The error ratios SER and TER decrease with the increase of $w$ on the data with BIG value. While the error ratios SER and TER increase with the increase of $w$ on the data with SMALL value. According to the results, if we want to accurately recover the data with big values, we should use a larger $w$. Otherwise, if we want to accurately recover the data with small values, we should use a smaller $w$.

### D. Comparison with Other Tensor Completion Algorithms

Besides ETC, we implement other four tensor completion algorithms CP-nmu [44], CP-opt [45], CP-als [4], and Tucker [46].

*1) Performance on The Data with BIG Value*

Fig.12(a) show the recovery performance on the data with BIG value with different sampling ratios. In this case, we focus on large data entries and set $w = 0.9$ in our ETC. As expected, with the increase of the sampling ratio thus sample data, TER decreases and thus better recovery performance is obtained. The recovery performance achieved by ETC is significantly better than the peer tensor completion algorithms. When the sampling ratio is low at 10%, the recovery error ratios on the large missing data is 0.2 using ETC, but is 0.7 using the reference tensor completion algorithms, which is 3.5 times larger. All these recovery performance results demonstrate that our ETC can achieve good recovery quality for data with big values.

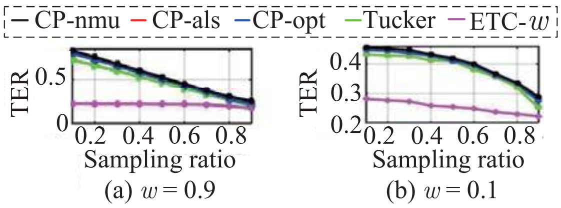



Figure 12. Comparison with different tensor completion algorithms.

From the probability theory and statistics, the cumulative distribution function (CDF) describes the probability for a real-valued random variable $X$ to have a value less than or equal to $x$, and $F(x) = p(X \le x)$. Therefore, to describe the distribution of individual relative error on the missing data, $\frac{|x_{ijk}-\hat{x}_{ijk}|}{x_{ijk}}\left((i,j,k)\in\bar{\Omega}\right)$, Fig.13(a) shows the CDF results of the individual error ratios under the sampling ratio of 30%. Obviously, ETC-0.9 has much better recovery performance and can achieve low error ratio with a high probability.

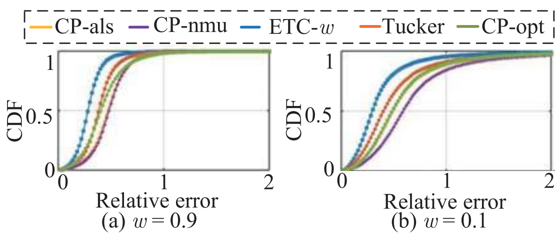



Figure 13. CDF of relative error via ETC with $w$=0.9 & $w$=0.1 on the data with BIG & SAMLL value

### 2) Performance on The Data with SMALL Value

Although this paper focuses on accurately recovering large entries as they are especially important to lots of network applications, we also implement experiments by setting $w = 0.1$ in our ETC to investigate the performance on the data with SMALL value. Fig.12(b), and Fig.13(b) show the experimental results. Obviously, our ETC-0.1 achieves the best recovery performance on the data with SMALL values as well.

These experimental results also demonstrate that our ETC has the ability of providing a flexible problem formulation for different focuses of the data recovery by setting different weight $w$.

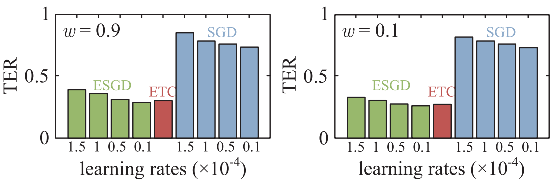
## E. Comparison with SGD (Stochastic Gradient Descent)

Besides our ETC, we implement two groups of SGD based tensor completion algorithms.

Group 1 (ESGD): The missing date recovery problem is the same as our expectile tensor completion problem in (12). We apply the SGD to solve the problem in (12).

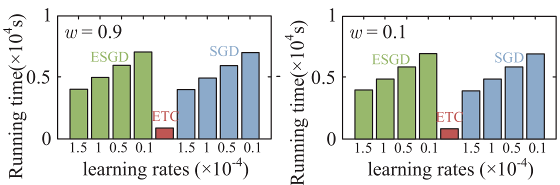
Group 2 (SGD): The missing data recovery problem is formulated in its conventional formulation in (5). We apply the SGD update rules in (7) to solve the problem in (5).

For data with BIG value and SMALL value, we set $w = 0.9$ and $w = 0.1$ in ESGD and our ETC, respectively. For ESGD and SGD, to investigate how the parameter learning rate impacts the recovery performance, we vary the learning rate from $0.1 \times 10^{-4}$ to $1.5 \times 10^{-4}$.




(a) BIG value (b) SMALL value

Figure 14. Recovery accuracy comparison with SGD approaches under different learning rates.




(a)BIG value (b)SMALL value

Figure 15. Speed comparison with SGD approaches under different learning rates.

Fig.14 show the recovery performance on the data with BIG value and SMALL value, respectively. As expected, ESGD and SGD achieve smaller error ratios thus higher recovery accuracy with a small learning rate. However, the small learning rate results in large computation cost and small recovery speed in Fig.15. When the learning rate is set to be very small ($0.1 \times 10^{-4}$), the accuracy of ESGD-$0.1 \times 10^{-4}$ is similar to those values of our ETC in Fig.14, respectively. However, our ETC's speeds are nearly 8 times that under ESGD-$0.1 \times 10^{-4}$ in Fig.15. These results demonstrate that our tensor completion algorithm in Section VI-A is very effective and efficient to obtain the optimal solution.

In Fig.14, we also observe that the error ratio TER under ESGD is much smaller than that under SGD. This is because that ESGD adopts expectile tensor completion problem formulation in (12) instead of the conventional formulation in (5). These results also demonstrate that our expectile regression enhanced formulation of tensor completion is general and effective to handle skewed data sets.

# VIII. Conclusion

Accurate inference of missing network traffic data is of great significance in network monitoring. Through experimental studies, we found network monitoring data are highly skewed with heavy tails. When we apply conventional tensor completion algorithms to these real datasets, we also found that the entries (especially the ones with large data values) inferred may have some large deviation from their true values. This is due to the conventional method of minimizing the mean square error tends to interpret the centrality of the data.

To address the issue, we apply expectile regression to the tensor completion and propose a novel problem formulation, expectile tensor completion, to better follow the data distribution for more accurate recovery of the large missing entries. It is an asymmetric least-square problem and can reflect both the centrality and the tail behavior of the data by setting different weights. Instead of ALS and SGD which are usually adopted to solve conventional tensor completion problem, we propose a simple yet efficient tensor completion algorithm. We further provide a detailed proof of the convergence of our algorithm by building an intelligent auxiliary function. We have done a great deal of experiments to assess the accuracy of ETC, and the results demonstrate that our algorithm is superior to the state of the art algorithms, especially in the recovery of extreme data. At the same time, our algorithm performs well even the sampling ratio is low.

# Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 62025201, Grant 61972144, and Grant 61976087, in part by the U.S. NSF under Grant ECCS 2030063, 1731238, CCF 2007313, in part by the CERNET Innovation Project under Grant NGII20190118, and in part by the Peng Cheng Laboratory Project of Guangdong Province under Grant PCL2018KP004.

# References

[1] P. Tune and M. Roughan. "Spatiotemporal traffic matrix synthesis," in ACM SIGCOMM CCR, vol. 45, pp. 579–592, ACM, 2015.
[2] I. Cunha, R. Teixeira, D. Veitch, and C. Diot. "Predicting and tracking internet path changes," in ACM SIGCOMM CCR, vol. 41, pp. 122–133, ACM, 2011.
[3] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu. "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," IEEE/ACM ToN, vol. 20, no. 3, pp. 662 – 676, 2012.
[4] K. Xie, C. Peng, X. Wang, G. Xie, J. Wen, J. Cao, D. Zhang, and Z. Qin. "Accurate recovery of internet traffic data under variable rate measurements," IEEE/ACM ToN, vol. 26, no. 3, pp. 1137–1150, 2018.

[5] 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.

[6] C. R. Kalmanek, S. Misra, and Y. R. Yang. "Guide to reliable internet services and applications," Springer Science Business Media, 2010.

[7] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, J. Cao, and D. Zhang. "Fast tensor factorization for accurate internet anomaly detection," IEEE/ACM ToN, vol. 25, no. 6, pp. 3794–3807, 2017.

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

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

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

[11] G. Gursun and M. Crovella. "On traffic matrix completion in the internet," in ACM IMC, 2012.

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

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

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

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

[16] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2013. "Tensor completion for estimating missing values in visual data," PAMI, 35, 1(2013), 208–220.

[17] Song Q , Ge H , Caverlee J. "Tensor Completion Algorithms in Big Data Analytics," ACM Trans. Knowl. Discov. Data 13, 1, Article 6 (January 2019).

[18] A. Sivaraman, S. Subramanian, M. Alizadeh, S. Chole, S.-T. Chuang, A. Agrawal, H. Balakrishnan, T. Edsall, S. Katti, and N. McKeown. "Programmable packet scheduling at line rate," in Proceedings of the 2016 ACM SIGCOMM Conference, ACM, 2016, pp. 44–57.

[19] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. "Deriving traffic demands for operational ip networks: Methodology and experience," in ACM SIGCOMM Computer Communication Review, vol. 30, no. 4. ACM, 2000, pp. 257–270.

[20] A. Lakhina, M. Crovella, and C. Diot. "Characterization of network-wide anomalies in traffic flows," in Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ACM, 2004, pp. 201–206.

[21] O. Rottenstreich and J. Tapolcai. "Optimal rule caching and lossy compression for longest prefix matching," IEEE/ACM Transactions on Networking (TON), vol. 25, no. 2, pp. 864–878, 2017.

[22] Zhang, Y., et al. "Spatio-temporal compressive sensing and internet traffic matrices," in ACM SIGCOMM Computer Communication Review, 2009. ACM.

[23] Y. Mao, L. Saul, and J. M. Smith. "IDES: An Internet distance estimation service for large networks," IEEE Journal On Selected Areas in Communications, vol. 24, no. 12, pp. 2273-2284, Dec, 2006.

[24] Liao, Y., et al. "DMFSGD: A decentralized matrix factorization algorithm for network distance prediction," IEEE/ACM Transactions on Networking, 2013. 21(5): p. 1511-1524.

[25] K. Xie, C. Peng, X. Wang, G. Xie, J. Wen. "Accurate Recovery of Internet Traffic Data UnderDynamic Measurements," in IEEE INFOCOM, 2017.

[26] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, G. Zhang. "Accurate Recovery of Internet Traffic Data: A Tensor Completion Approach," in IEEE INFOCOM, 2016.

[27] K. Xie, X.G. Wang, X. Wang, Y. X. Chen. "Accurate Recovery of Missing Network Measurement Data With Localized Tensor Completion," IEEE/ACM ToN, vol. 27, no. 6, pp. 2222-2235, 2019.

[28] K. Xie, H. Hua, X. Wang, G. Xie. "Neural Tensor Completion for Accurate Network Monitoring," in IEEE INFOCOM, 2020.

[29] Y. Liu, Z. Long, H. Huang, and C. Zhu. "Low CP rank and Tucker rank tensor completion for estimating missing components in image data," IEEE TCSVT, DOI: 10.1109/TCSVT.2019.2901311, 2019.

[30] L. Zhang, L. Song, B. Du and Y. Zhang. "Nonlocal LowRank Tensor Completion for Visual Data," IEEE TCYB, doi: 10.1109/TCYB.2019.2910151.

[31] Z. Long, Y. Liu, L. Chen, and C. Zhu. "Low rank tensor completion for multiway visual data," SP, vol. 155, pp. 301–316, 2019.

[32] M. Ashraphijuo and X. Wang. "Fundamental conditions for low-cp-rank tensor completion," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2116–2145, 2017.

[33] Q. Zhao, L. Zhang, and A. Cichocki. "Bayesian CP factorization of incomplete tensors with automatic rank determination," IEEE TPAMI, vol. 37, no. 9, pp. 1751–1763, 2015.

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

[35] L. De Lathauwer, "Blind separation of exponential polynomials and the decomposition of a tensor in rank-(l_r,l_r,1) terms," SIAM Journal on Matrix Analysis and Applications, vol. 32, no. 4, pp. 1451–1474, 2011.

[36] B. W. Bader and T. G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," ACM Transactions on Mathematical Software (TOMS), vol. 32, no. 4, pp. 635–653, 2006

[37] X. He, H. Zhang, M. Y. Kan, and T. S. Chua. "Fast matrix factorization for online recommendation with implicit feedback," in Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 549–558, 2016.

[38] R. Zhu, D. Niu and Z. Li. "Robust web service recommendation via quantile matrix factorization," in Proceedings of IEEE INFOCOM 2017

[39] Koenker, R., and Bassett Jr, G. "Regression quantiles," Econometrica, 33–50, 1978.

[40] Newey, W. K., and Powell, J. L. "Asymmetric least squares estimation and testing," Econometrica, 819–847, 1987.

[41] R. Zhu, D. Niu, L. Kong, and Z. Li. "Expectile Matrix Factorization for Skewed Data Analysis," in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA

[42] The Abilene Observatory Data Collections. Accessed: Jul. 20, 2004. [Online]. Available http://abilene.internet2.edu/observatory/datacollections.html.

[43] 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.

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

[45] E. Acar and T. G. Dunlavy, Daniel M.and Kolda. "A scalable optimization approach for fitting canonical tensor decompositions," J. Chemom, vol. 25, no. 2, p. 6786, 2011.

[46] B. W.Bader, T. G.Kolda, et al. "Matlab tensor toolbox version 2.5," Available online, January 2012. [Online]. Available: http://www.sandia.gov/ tgkolda/TensorToolbox/.