Scalable Low-Rank Autoregressive Tensor Learning for Spatiotemporal Traffic Data Imputation

Xinyu Chen[a], Yixian Chen[b], Lijun Sun[a,c,*]

[a]*Department of Civil Engineering, McGill University, Montreal, QC H3A 0C3, Canada*
[b]*School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou 510006, China*
[c]*Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, QC H3T 1J4, Canada*

Abstract

Missing value problem in spatiotemporal traffic data has long been a challenging topic, in particular for large-scale and high-dimensional data with complex missing mechanisms and diverse degrees of missingness. Recent studies based on tensor nuclear norm have demonstrated the superiority of tensor learning in imputation tasks by effectively characterizing the complex correlations/dependencies in spatiotemporal data. However, despite the promising results, these approaches do not scale well to large tensors. In this paper, we focus on addressing the missing data imputation problem for large-scale spatiotemporal traffic data. To achieve both high accuracy and efficiency, we develop a scalable autoregressive tensor learning model—Low-Tubal-Rank Autoregressive Tensor Completion (LATC-Tubal)—based on the existing framework of Low-Rank Autoregressive Tensor Completion (LATC by Chen and Sun (2020)), which is well-suited for spatiotemporal traffic data that characterized by multidimensional structure of location$\times$ time of day $\times$ day. In particular, the proposed LATC-Tubal model involves a scalable tensor nuclear norm minimization scheme by integrating linear unitary transformation. Therefore, the tensor nuclear norm minimization can be solved by singular value thresholding on the transformed matrix of each day while the day-to-day correlation can be effectively preserved by the unitary transform matrix. Before setting up the experiment, we consider two large-scale 5-minute traffic speed data sets collected by the California PeMS system with 11160 sensors: 1) PeMS-4W covers the data over 4 weeks (i.e., 288×28 time points), and 2) PeMS-8W covers the data over 8 weeks (i.e., 288×56 time points). We compare LATC-Tubal with state-of-the-art baseline models, and find that LATC-Tubal can achieve competitively accuracy with a significantly lower computational cost. In addition, the LATC-Tubal will also benefit other tasks in modeling large-scale spatiotemporal traffic data, such as network-level traffic forecasting.

Keywords: Spatiotemporal traffic data, High-dimensional data, Missing data imputation, Low-rank tensor completion, Autoregressive process, Linear unitary transformation

1. Introduction

With remarkable advances in low-cost sensing technologies, an increasing amount of real-world traffic measurement data are collected with high spatial sensor coverage and fine temporal resolution. These large-scale and high-dimensional traffic data provide us with unprecedented opportunities for sensing traffic dynamics and developing efficient and reliable applications for intelligent transportation systems (ITS). However, there are two critical issues that undermine the use of these data in real-world applications: (1) the presence of missing and corrupted data remains a primary challenge and makes it difficult to get the true signals, and (2) it is computationally challenging to perform large-scale and high-dimensional traffic data analysis using existing methods/algorithms. The issue of missing and corrupted data may arise from complicated sensor malfunctioning, communication failure, and even unsatisfied sensor coverage. To improve data quality and support downstream applications, missing data imputation becomes an essential task. Despite recent efforts and advances in developing machine learning-based imputation models, most studies still focus on small-scale data sets (e.g., 100~400 time series) and it remains a critical challenge to perform accurate and efficient imputation on large-scale traffic data sets.

In the literature, there are numerous methods for modeling corrupted spatiotemporal traffic data. We summarize these methods from the aspect of time series modeling as the following two main categories. The first category focuses on single time series modeling, which includes regression methods such as linear regression and autoregressive model (Schafer, 1997; Chen and Shao, 2000). Although this approach is simple, it can not characterize the correlations/dependencies among different sensors in a spatiotemporal setting. To address

---

this problem, the second category focuses on multivariate/multidimensional setting by modeling multivariate time series data using matrix/tensor structures. In this category, signature frameworks include: 1) classical time series analysis methods like vector autoregressive model (Bashir and Wei, 2018), 2) purely low-rank matrix factorization/completion, e.g., low-rank matrix factorization (Asif et al., 2013, 2016), principle component analysis (Qu et al., 2008, 2009) and their variants, 3) purely low-rank tensor factorization/completion, e.g., Bayesian tensor factorization (Chen et al., 2019) and low-rank tensor completion (Ran et al., 2016; Chen et al., 2020), 4) low-rank matrix factorization by integrating time series models, e.g., temporal regularized matrix factorization (Yu et al., 2016) and Bayesian temporal matrix factorization (Sun and Chen, 2019), and 5) low-rank tensor completion by integrating time series models like LATC (Chen and Sun, 2020). For this large family of methods, a common goal is to capture both temporal dynamics and spatial consistency. For example, by integrating time series models and purely low-rank tensor completion models, the recently developed LATC framework has shown superior performance compared with existing low-rank matrix/tensor completion models (Chen and Sun, 2020) by imposing both local consistency and global consistency in traffic series data (Li et al., 2015). In addition, there are also some nonlinear methods (e.g., deep learning in Che et al. (2018)) that have been applied to address the problem of missing traffic data. Nevertheless, how to generalize these methods to large-scale problems is a long standing technical gap between research and real-world applications.

Thanks to recent advances in low-rank model, it become technically feasible to solve large-scale matrix/tensor learning problems through a series of "small" subproblems by integrating linear transforms (e.g., discrete Fourier transform, discrete cosine transform) (Rao and Yip, 1990). For instance, to learn from partially observed tensor data in large-scale image/video data, recent studies have invertible linear transformation into LRTC. Linear transforms like Fourier transform and wavelet transform serve a key role in developing accurate and fast multidimensional tensor completion models (Kernfeld et al., 2015). Lu et al. (2016, 2019); Lu et al. (2020) introduced some invertible linear transforms (e.g., discrete Fourier transform, discrete cosine transform) into LRTC by connecting tensor tubal rank with tensor singular value decomposition (SVD). As demonstrated in these work, LRTC with invertible linear transforms outperforms by a large margin - LRTC with weighted tensor nuclear norm proposed by Liu et al. (2013). The discrete Fourier transform is the most commonly-used invertible linear transform in the relevant work. To enhance LRTC algorithms with well-suited linear transform, Song et al. (2020) tried to integrate a data-driven unitary transform (computed from singular vectors of data) into LRTC. In their work, they concluded from numerical experiments that unitary transform performs better than both discrete Fourier transform and discrete wavelet transform for tensor completion.

In modeling spatiotemporal traffic data, the most fundamental assumption is that the data has an inherent low-rank structure either with the matrix or with the tensor representations. Therefore, the idea of combining tensor completion and linear transform is also well-suited for spatiotemporal traffic data because day-to-day or week-to-week correlation of the multivariate and high-dimensional traffic time series can be effectively encoded by linear transforms. Inspired by the idea of linear transforms, we can first convert large-scale tensor completion into a series of 'small' daily or weekly traffic data imputation problems and then connect their solutions by linear transforms, thus overcoming the scalability challenge. Following this idea, in this work, we introduce a scalable low-rank autoregressive tensor learning model to missing traffic data imputation. Our contribution is three-fold:

- We develop a Low-Tubal-Rank Autoregressive Tensor Completion (LATC-Tubal) model by integrating linear unitary transforms, which is both accurate and efficient for large-scale and high-dimensional traffic data imputation.

- We provide new insight on spatiotemporal data modeling where the large-scale data tensor problem can be solved equivalently through a series of "small" subproblems by introducing linear transform.

- We conduct extensive imputation experiments on two real-world large-scale traffic data sets. The results show that the proposed LATC-Tubal model is much more efficient than state-of-the-art models while maintaining comparable accuracy.

The remainder of this paper is organized as follows. We introduce notations and problem definition in Section 2. Section 3 introduces in detail the proposed low-rank autoregressive tensor learning model. In Section 4, we conduct numerical experiments on large-scale traffic data sets and show the performance of LATC-Tubal compared with several state-of-the-art imputation models. Finally, we summarize the study and discuss future research directions in Section 5.

2. Preliminaries

2.1. Notations

In this work, we use boldface uppercase letters to denote matrices, e.g., $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, boldface lowercase letters to denote vectors, e.g., $\boldsymbol{x} \in \mathbb{R}^M$, and lowercase letters to denote scalars, e.g., x. Given a matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, we use $\boldsymbol{X}_{[:t]} \in \mathbb{R}^{M \times t}$ and $\boldsymbol{X}_{[t+1:]} \in \mathbb{R}^{M \times (N-t)}$ to denote the submatrices of $\boldsymbol{X}$ that consist of the first t columns and the
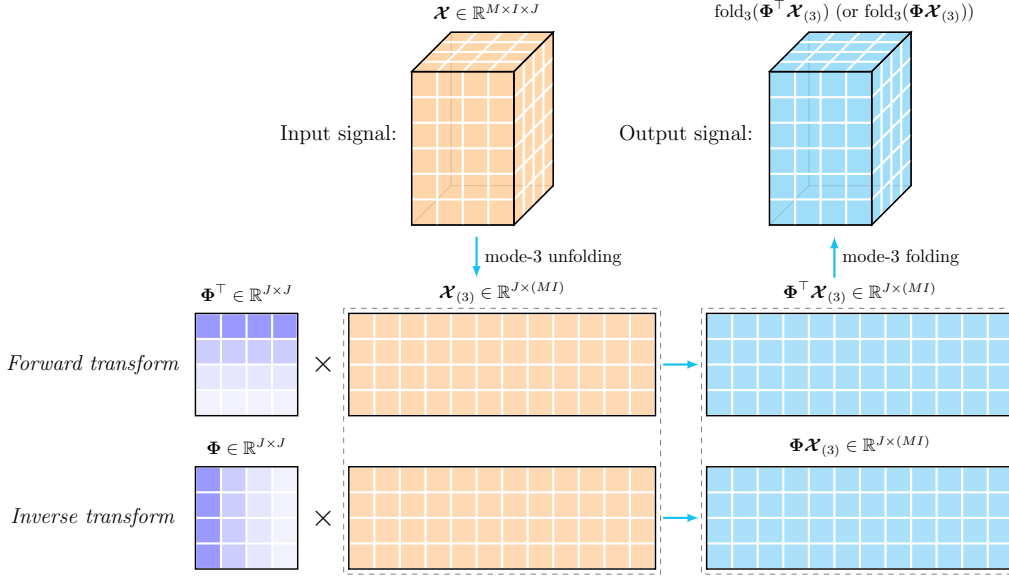
Figure 1: Schematic of the forward and inverse unitary transforms along the third dimension of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times I \times J}$ where $\boldsymbol{\Phi} \in \mathbb{R}^{J \times J}$ is the unitary transform matrix.

last $N - t$ columns, respectively. We denote the (m, n)th entry in $\boldsymbol{X}$ by $x_{m,n}$. The Frobenius norm of $\boldsymbol{X}$ is defined as $\|\boldsymbol{X}\|_F = \sqrt{\sum_{m,n} x_{m,n}^2}$, and the ℓ_2-norm of $\boldsymbol{x}$ is defined as $\|\boldsymbol{x}\|_2 = \sqrt{\sum_m x_m^2}$. We denote a third-order tensor by $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times I \times J}$ and the kth-mode ($k = 1, 2, 3$) unfolding of $\boldsymbol{\mathcal{X}}$ by $\boldsymbol{\mathcal{X}}_{(k)}$ (Kolda and Bader, 2009). Correspondingly, we define a folding operator that converts a matrix to a third-order tensor in the kth-mode as $\text{fold}_k(\cdot)$; thus, we have $\text{fold}_k(\boldsymbol{\mathcal{X}}_{(k)}) = \boldsymbol{\mathcal{X}}$. For the tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times I \times J}$, we define its Frobenius norm as $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\sum_{m,i,j} x_{m,i,j}^2}$ and its inner product with another tensor as $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{m,i,j} x_{mij} y_{mij}$ where $\boldsymbol{\mathcal{Y}}$ shares the same size with $\boldsymbol{\mathcal{X}}$.

Unitary transform, as a powerful tool in the digital signal processing area, is defined as the invertible linear transform which includes discrete cosine/sine transform and others satisfying the rules of orthogonality and normalization (Rao and Yip, 1990). We denote by $\boldsymbol{\Phi}[\cdot]$ the unitary transform. Without loss of generality, we specify that the unitary transform imposed on any tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times I \times J}$ is along the third dimension and assume that unitary transform is acted on real-valued data. Therefore, the forward unitary transform can be defined as follows,

$$\boldsymbol{\Phi}[\boldsymbol{\mathcal{X}}] \equiv \text{fold}_3(\boldsymbol{\Phi}^\top \boldsymbol{\mathcal{X}}_{(3)}) \in \mathbb{R}^{M \times I \times J}, \tag{1}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{J \times J}$ is the unitary matrix. Correspondingly, we define the inverse unitary transform for any given tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times I \times J}$ as

$$\boldsymbol{\Phi}^{-1}[\boldsymbol{\mathcal{X}}] \equiv \text{fold}_3(\boldsymbol{\Phi} \boldsymbol{\mathcal{X}}_{(3)}) \in \mathbb{R}^{M \times I \times J}. \tag{2}$$

Putting the forward and inverse unitary transforms together, we have $\boldsymbol{\mathcal{X}} = \boldsymbol{\Phi}^{-1}[\boldsymbol{\Phi}[\boldsymbol{\mathcal{X}}]]$. Figure 1 intuitively shows how to implement forward and inverse unitary transforms along the third dimension of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{M \times I \times J}$. In this work, we generate the unitary matrix $\boldsymbol{\Phi}$ by the SVD of mode-3 unfolding of the data tensor, and Song et al. (2020) called the derived matrix as data-driven unitary transform matrix.

2.2. Problem Definition

In this work, we introduce spatiotemporal missing time series data imputation tasks in a general sense. For any given partially observed data matrix $\boldsymbol{Y}$ whose columns correspond to time points and rows correspond to spatial locations/sensors:

$$\boldsymbol{Y} = \left[\begin{array}{ccccc} | & | & & | \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_{IJ} \\ | & | & & | \end{array} \right] \in \mathbb{R}^{M \times (IJ)}, \tag{3}$$

where M is the number of spatial locations/sensors and IJ is the number of continuous time points. Then the observed values of $\boldsymbol{Y}$ can be written as $\mathcal{P}_\Omega(\boldsymbol{Y})$ in which the operator $\mathcal{P}_\Omega : \mathbb{R}^{M \times (IJ)} \mapsto \mathbb{R}^{M \times (IJ)}$ is an orthogonal projection supported on the observed index set Ω:

$$[\mathcal{P}_\Omega(\boldsymbol{Y})]_{m,n} = \begin{cases} y_{m,n}, & \text{if } (m, n) \in \Omega, \\ 0, & \text{otherwise}, \end{cases}$$

where $m = 1, ..., M$ and $n = 1, ..., IJ$. The modeling target of missing data imputation can be described as learning the unobserved values from partially observed $\mathcal{P}_\Omega(\boldsymbol{Y})$.
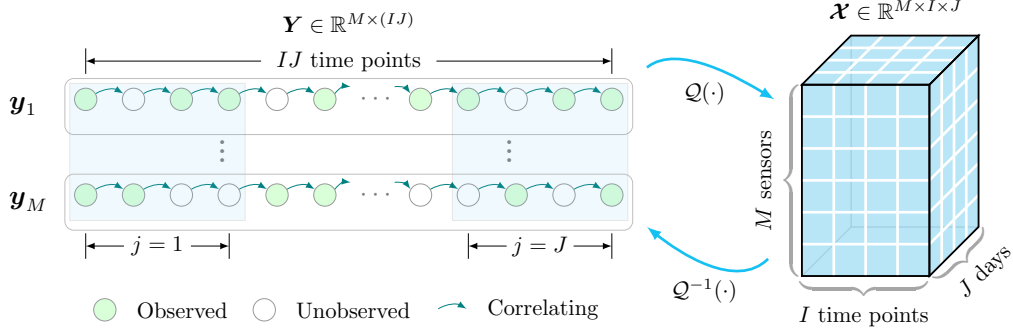
3

Figure 2: Illustration of the proposed LATC framework for spatiotemporal traffic data imputation.

In particular, to take advantage of the algebraic structure of tensor, we introduce a forward tensorization operator $\mathcal{Q}(\cdot)$ that converts the spatiotemporal time series matrix into a third-order tensor (see Figure 2). This operation is implemented by splitting the time dimension as (time point per day, day)-indexed combinations. For example, we can generate a third-order tensor by the forward tensorization operator as $\boldsymbol{\mathcal{X}} = \mathcal{Q}(\boldsymbol{Y}) \in \mathbb{R}^{M \times I \times J}$. Conversely, we can convert the resulted tensor into the original matrix by $\boldsymbol{Y} = \mathcal{Q}^{-1}(\boldsymbol{\mathcal{X}}) \in \mathbb{R}^{M \times (IJ)}$ in which $\mathcal{Q}^{-1}(\cdot)$ denotes the inverse operator of $\mathcal{Q}(\cdot)$.

3. Low-Rank Autoregressive Tensor Learning

In this section, we develop a Low-Tubal-Rank Autoregressive Tensor Completion (LATC-Tubal) model that relies on the Low-Rank Autoregressive Tensor Completion (LATC) framework proposed by Chen and Sun (2020). To overcome the large scale and high dimensionality of the real-world traffic data, we propose to integrate linear unitary transform into the LATC-Tubal model to reduce the computational cost of the tensor singular value thresholding process.

3.1. Model Description

In fact, spatiotemporal traffic data are multivariate time series associated with both spatial and temporal dimensions. To characterize these data with a suitable algebraic structure, we could represent these data with multivariate time series matrix or multidimensional tensor. Thus, the motivation behind modeling spatiotemporal traffic data in this work is to build an autoregressive process on the data matrix and impose a low-rank structure on data tensor simultaneously. To do so, the essential idea of LATC-Tubal takes into account both tensor nuclear norm minimization and autoregression error/loss minimization in a unified LATC framework (Chen and Sun, 2020):

$$\min_{\boldsymbol{\mathcal{X}}, \boldsymbol{Z}, \boldsymbol{A}} \ \|\boldsymbol{\mathcal{X}}\|_* + \lambda \|\boldsymbol{Z}\|_{\boldsymbol{A}, \mathcal{H}}$$
$$\text{s.t.} \ \begin{cases} \boldsymbol{\mathcal{X}} = \mathcal{Q}(\boldsymbol{Z}), \\ \mathcal{P}_\Omega(\boldsymbol{Z}) = \mathcal{P}_\Omega(\boldsymbol{Y}), \end{cases} \tag{4}$$

where $\boldsymbol{Y} \in \mathbb{R}^{M \times (IJ)}$ is the partially observed time series matrix. In the objective of Eq. (4), we strengthen the existing LATC model by providing a rigorous connection between tensor tubal rank and tensor nuclear norm (denote by $\|\cdot\|_*$) through invertible linear transform. Notably, under invertible linear transform, tensor nuclear norm minimization has been verified as the convex surrogate to the tensor tubal rank minimization (Kernfeld et al., 2015; Lu et al., 2016); $\|\cdot\|_{\boldsymbol{A}, \mathcal{H}}$ denotes the autoregressive norm of multivariate time series matrix which holds the following definition:

$$\|\boldsymbol{Z}\|_{\boldsymbol{A}, \mathcal{H}} = \sum_{m, t} (z_{m, t} - \sum_i a_{m, i} z_{m, t - h_i})^2, \tag{5}$$

for matrix $\boldsymbol{Z}$ with a lag set $\mathcal{H}$ (i.e., $\mathcal{H} = \{h_1, ..., h_d\}$) and coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{M \times d}$. The autoregression loss consists in the autoregression loss of each independent time series. Note that, with this definition, we also need to estimate the variable $\boldsymbol{A}$. The parameter λ is the weight parameter that controls the trade-off between the tensor nuclear norm and the autoregressive norm.

In the above formulation, $\boldsymbol{\mathcal{X}}$ is the low-rank tensor variable associated with tensor nuclear norm minimization, and $\boldsymbol{Z}$ is the time series matrix variable characterized by autoregressive process. We can see intuitively from Figure 2 that these two variables are able to recover $\boldsymbol{Y}$ with time series dynamics and low-rank properties because the established constraint in Eq. (4), i.e., $\boldsymbol{\mathcal{X}} = \mathcal{Q}(\boldsymbol{Z})$, is closely related to the partially observed matrix $\boldsymbol{Y}$.

4

To resolve the optimization problem as given in Eq. (4), we use the Alternating Direction Method of Multipliers (ADMM) algorithm whose augmented Lagrangian function associated with the problem (4) is defined as follows,

$$\mathcal{L}(\boldsymbol{\mathcal{X}}, \boldsymbol{Z}, \boldsymbol{A}, \boldsymbol{\mathcal{T}}) = \|\boldsymbol{\mathcal{X}}\|_* + \lambda\|\boldsymbol{Z}\|_{\boldsymbol{A},\mathcal{H}} + \frac{\rho}{2}\|\boldsymbol{\mathcal{X}} - \mathcal{Q}(\boldsymbol{Z})\|_F^2 + \langle \boldsymbol{\mathcal{X}} - \mathcal{Q}(\boldsymbol{Z}), \boldsymbol{\mathcal{T}} \rangle, \tag{6}$$

where ρ is the learning rate of ADMM algorithm, $\langle \cdot, \cdot \rangle$ denotes the inner product and $\boldsymbol{\mathcal{T}}$ is the dual variable. In particular, $\mathcal{P}_\Omega(\boldsymbol{Z}) = \mathcal{P}_\Omega(\boldsymbol{Y})$ is the fixed constraint for maintaining observation information.

According to the augmented Lagrangian function in Eq. (6), ADMM transforms the original tensor completion problem into the following subproblems in an iterative manner:

$$\boldsymbol{\mathcal{X}}^{\ell+1} := \arg\min_{\boldsymbol{\mathcal{X}}} \mathcal{L}(\boldsymbol{\mathcal{X}}, \boldsymbol{Z}^\ell, \boldsymbol{A}^\ell, \boldsymbol{\mathcal{T}}^\ell), \tag{7}$$

$$\boldsymbol{Z}^{\ell+1} := \arg\min_{\boldsymbol{Z}} \mathcal{L}(\boldsymbol{\mathcal{X}}^{\ell+1}, \boldsymbol{Z}, \boldsymbol{A}^\ell, \boldsymbol{\mathcal{T}}^\ell), \tag{8}$$

$$\boldsymbol{A}^{\ell+1} := \arg\min_{\boldsymbol{A}} \mathcal{L}(\boldsymbol{\mathcal{X}}^{\ell+1}, \boldsymbol{Z}^{\ell+1}, \boldsymbol{A}, \boldsymbol{\mathcal{T}}^\ell), \tag{9}$$

$$\boldsymbol{\mathcal{T}}^{\ell+1} := \boldsymbol{\mathcal{T}}^\ell + \rho(\boldsymbol{\mathcal{X}}^{\ell+1} - \mathcal{Q}(\boldsymbol{Z}^{\ell+1})), \tag{10}$$

where ℓ denotes the count of iteration. Note that the estimates of these variables can be solved recursively until convergence. Eq. (10) shows the dual update in this ADMM. In what follows, we will discuss and present the detailed solutions of these subproblems (7), (8), and (9).

3.2. Computing the Variable $\boldsymbol{\mathcal{X}}$

With respect to variable $\boldsymbol{\mathcal{X}}$, the optimization expression in Eq. (6) can be simplified to a standard tensor nuclear norm minimization. Thus, Eq. (7) yields

$$\begin{aligned}
\boldsymbol{\mathcal{X}}^{\ell+1} := &\arg\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{X}}\|_* + \frac{\rho}{2}\|\boldsymbol{\mathcal{X}} - \mathcal{Q}(\boldsymbol{Z}^\ell)\|_F^2 + \langle \boldsymbol{\mathcal{X}} - \mathcal{Q}(\boldsymbol{Z}^\ell), \boldsymbol{\mathcal{T}}^\ell \rangle \\
= &\arg\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{X}}\|_* + \frac{\rho}{2}\|\boldsymbol{\mathcal{X}} - (\mathcal{Q}(\boldsymbol{Z}^\ell) - \boldsymbol{\mathcal{T}}^\ell/\rho)\|_F^2,
\end{aligned} \tag{11}$$

where this optimization can be carried out as a tensor singular value thresholding.

Lemma 1. *(Song et al., 2020) For any tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{M \times I \times J}$, suppose that $\boldsymbol{\Phi} \in \mathbb{R}^{J \times J}$ is a unitary transform matrix, and the SVD of the unitary transformed matrix $\boldsymbol{\Phi}_j[\boldsymbol{\mathcal{Z}}]$ (i.e., the jth frontal slice of the unitary transformed tensor $\boldsymbol{\Phi}[\boldsymbol{\mathcal{Z}}]$) is*

$$\boldsymbol{\Phi}_j[\boldsymbol{\mathcal{Z}}] = \boldsymbol{U}_{\boldsymbol{\Phi}} \boldsymbol{\Sigma}_{\boldsymbol{\Phi}} \boldsymbol{V}_{\boldsymbol{\Phi}}^\top, \tag{12}$$

where $\boldsymbol{U}_{\boldsymbol{\Phi}} \in \mathbb{R}^{M \times R}, \boldsymbol{\Sigma}_{\boldsymbol{\Phi}} \in \mathbb{R}^{R \times R}, \boldsymbol{V}_{\boldsymbol{\Phi}} \in \mathbb{R}^{I \times R}$ ($R = \min\{M, I\}$). Then an optimal solution to the following optimization problem

$$\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{X}}\|_* + \frac{\rho}{2}\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}}\|_F^2, \tag{13}$$

is given by the tensor singular value thresholding:

$$\hat{\boldsymbol{X}}_j := \boldsymbol{\Phi}^{-1}[\mathcal{D}_{1/\rho}(\boldsymbol{\Phi}_j[\boldsymbol{\mathcal{Z}}])], \quad \mathcal{D}_{1/\rho}(\boldsymbol{\Phi}_j[\boldsymbol{\mathcal{Z}}]) = \boldsymbol{U}_{\boldsymbol{\Phi}}[\boldsymbol{\Sigma}_{\boldsymbol{\Phi}} - 1/\rho]_+ \boldsymbol{V}_{\boldsymbol{\Phi}}^\top, \tag{14}$$

where matrix $\hat{\boldsymbol{X}}_j \in \mathbb{R}^{M \times I}$ is the jth frontal slice of the estimated tensor $\hat{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{M \times I \times J}$, and $\mathcal{D}_{1/\rho}(\cdot)$ denotes the operator of matrix singular value thresholding associated with the parameter ρ.

In this lemma, the tensor singular value thresholding of any given tensor is achieved by first applying matrix singular value thresholding to the forward unitary transformed matrices, and then stacking the inverse unitary transformed results of matrix singular value thresholding to the slices of the estimated tensor. It is not difficult to see that if we impose a unitary transform along tensor's dimension, then the tensor nuclear norm minimization problem can be performed in a computationally efficient manner by solving nuclear norm minimization problems of 'small' matrix, making it scalable to high-dimensional data. In the literature, invertible linear transforms applied to the tensor completion problem have also been proved to satisfy certain tensor operation property (Kernfeld et al., 2015; Lu et al., 2016).

As shown in Figure 3, we provide an intuitive understanding of Lemma 1. We can see that the tensor singular value thresholding is well-suited for spatiotemporal traffic data that characterized by the dimensions of "sensor", "time point of day", and "day". The corresponding subproblems in the form of matrix allow one to implement singular value thresholding on the unitary transformed matrix for each day while the day-to-day correlation can be effectively preserved by the unitary transform matrix.

Therefore, regarding the problem (11), one could get the closed-form solution by using the tensor singular value thresholding as mentioned above:

$$\boldsymbol{X}_j^{\ell+1} := \boldsymbol{\Phi}^{-1}[\mathcal{D}_{1/\rho}(\boldsymbol{\Phi}_j[\mathcal{Q}(\boldsymbol{Z}^\ell) - \boldsymbol{\mathcal{T}}^\ell/\rho])], \tag{15}$$

where $\boldsymbol{X}_j^{\ell+1}, j = 1, ..., J$ are the slices of the tensor $\boldsymbol{\mathcal{X}}^{\ell+1}$. Note that this solution follows Eq. (14) in Lemma 1 directly. To get a matrix form solution, we compute $\hat{\boldsymbol{X}} = \mathcal{Q}^{-1}(\boldsymbol{\mathcal{X}}^{\ell+1}) \in \mathbb{R}^{M \times (IJ)}$ where $\boldsymbol{\mathcal{X}}^{\ell+1}$ is stacked by $\boldsymbol{X}_j^{\ell+1}$.
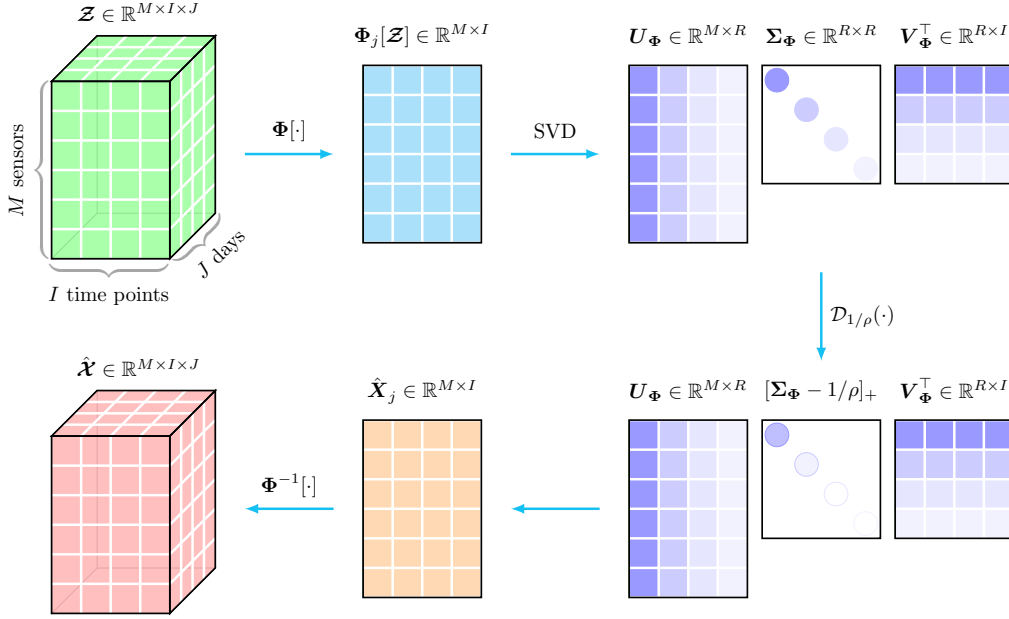
Figure 3: Schematic of tensor singular value thresholding ($R = \min\{M, I\}$) with unitary transform, in which $\boldsymbol{\Phi}[\cdot]$ and $\boldsymbol{\Phi}^{-1}[\cdot]$ denote the forward unitary transform and inverse unitary transform, respectively.

3.3. Computing the Variable $\boldsymbol{Z}$

For the fact that $\boldsymbol{\mathcal{X}} = \mathcal{Q}(\boldsymbol{Z})$ shown in Eq. (4), we can rewrite Eq. (7) with respect to the variable $\boldsymbol{Z}$ as follows by putting Eq. (6) and (8) together:

$$
\begin{aligned}
\boldsymbol{Z}^{\ell+1} := \arg\min_{\boldsymbol{Z}} \ & \lambda\|\boldsymbol{Z}\|_{\boldsymbol{A},\mathcal{H}} + \frac{\rho}{2}\|\boldsymbol{\mathcal{X}}^{\ell+1} - \mathcal{Q}(\boldsymbol{Z})\|_F^2 - \langle \mathcal{Q}(\boldsymbol{Z}), \boldsymbol{\mathcal{T}}^\ell \rangle \\
= \arg\min_{\boldsymbol{Z}} \ & \lambda \sum_m \|\boldsymbol{z}_{m,[h_d+1:]} - \boldsymbol{V}_m \boldsymbol{a}_m\|_2^2 + \frac{\rho}{2}\|\boldsymbol{\mathcal{X}}^{\ell+1} - \mathcal{Q}(\boldsymbol{Z})\|_F^2 - \langle \mathcal{Q}(\boldsymbol{Z}), \boldsymbol{\mathcal{T}}^\ell \rangle,
\end{aligned}
\tag{16}
$$

where $\boldsymbol{V}_m = (\boldsymbol{v}_{h_d+1}, \cdots, \boldsymbol{v}_{IJ})^\top \in \mathbb{R}^{(IJ-h_d)\times d}$ is an auxiliary matrix with $\boldsymbol{v}_t = (\hat{x}_{m,t-h_1}, \cdots, \hat{x}_{m,t-h_d})^\top \in \mathbb{R}^d, \forall t \in \{h_d + 1, ..., IJ\}$ formed by $\hat{\boldsymbol{X}}$.

According to the formula of Eq. (16), we need to separate this optimization problem as two subproblems:

$$
\begin{aligned}
\boldsymbol{Z}^{\ell+1}_{[:h_d]} := \arg\min_{\boldsymbol{Z}} \ & \frac{\rho}{2}\|\mathcal{Q}^{-1}(\boldsymbol{\mathcal{X}}^{\ell+1})_{[:h_d]} - \boldsymbol{Z}\|_F^2 - \langle \boldsymbol{Z}, \mathcal{Q}^{-1}(\boldsymbol{\mathcal{T}}^\ell)_{[:h_d]} \rangle, \\
\boldsymbol{z}^{\ell+1}_{m,[h_d+1:]} := \arg\min_{\boldsymbol{z}} \ & \lambda\|\boldsymbol{z} - \boldsymbol{V}_m \boldsymbol{a}_m^\ell\|_2^2 + \frac{\rho}{2}\|\mathcal{Q}^{-1}(\boldsymbol{\mathcal{X}}^{\ell+1})_{m,[h_d+1:]} - \boldsymbol{z}\|_2^2 - \langle \boldsymbol{z}, \mathcal{Q}^{-1}(\boldsymbol{\mathcal{T}}^\ell)_{m,[h_d+1:]} \rangle,
\end{aligned}
\tag{17}
$$

where for each problem, the optimization has a closed-form least-square solution. They are given by

$$
\boldsymbol{Z}^{\ell+1}_{[:h_d]} := \mathcal{Q}^{-1}(\boldsymbol{\mathcal{X}}^{\ell+1} + \boldsymbol{\mathcal{T}}^\ell/\rho)_{[:h_d]},
\tag{18}
$$

and

$$
\boldsymbol{z}^{\ell+1}_{m,[h_d+1:]} := \frac{\lambda}{\rho+\lambda}\boldsymbol{V}_m \boldsymbol{a}_m^\ell + \frac{1}{\rho+\lambda}\mathcal{Q}^{-1}(\rho\boldsymbol{\mathcal{X}}^{\ell+1} + \boldsymbol{\mathcal{T}}^\ell)_{m,[h_d+1:]},
\tag{19}
$$

respectively. The significance of these results is that we can preserve the time series dynamics by the variable $\boldsymbol{Z}^{\ell+1}$. By combining the low-rank patterns captured by the variable $\boldsymbol{\mathcal{X}}^{\ell+1}$, we can therefore get the estimated results involving both time series dynamics and low-rank patterns.

3.4. Computing the Variable $\boldsymbol{A}$

As mentioned above, the variable $\boldsymbol{A} \in \mathbb{R}^{M\times d}$ is the coefficient matrix in the autoregressive part. To estimate these parameters, we can solve the following minimization problem derived from Eq. (9):

$$
\begin{aligned}
\boldsymbol{A}^{\ell+1} := \arg\min_{\boldsymbol{A}} \ & \sum_{m,t}\Big(z_{m,t} - \sum_i a_{m,i}z_{m,t-h_i}\Big)^2 \\
\simeq \arg\min_{\boldsymbol{A}} \ & \sum_m \|\boldsymbol{z}_{m,[h_d+1:]} - \boldsymbol{V}_m \boldsymbol{a}_m\|_2^2,
\end{aligned}
\tag{20}
$$

6

where $\boldsymbol{V}_m$ is already defined in Eq. (16). It is not difficult to see that this optimization has a least-square solution, which is given by

$$\boldsymbol{a}_m^{\ell+1} := \boldsymbol{V}_m^+ \boldsymbol{z}_{m,[h_d+1:]}, \forall m, \tag{21}$$

where $\cdot^+$ denotes the pseudo-inverse of a matrix.

In this work, we devise a compact sampling process when implementing Eq. (21). Specifically, we first randomly sample certain amount of rows from the $IJ - h_d$ rows of $\boldsymbol{V}_m$ to construct $\tilde{\boldsymbol{V}}_m \in \mathbb{R}^{|\Psi| \times d}$ where we denote by Ψ ($|\Psi| \ll IJ - h_d$) the row index. Then we construct $\tilde{\boldsymbol{z}}_{m,[h_d+1:]} \in \mathbb{R}^{|\Psi|}$ accordingly. To do so, we can get the coefficient vectors by

$$\boldsymbol{a}_m^{\ell+1} := \tilde{\boldsymbol{V}}_m^+ \tilde{\boldsymbol{z}}_{m,[h_d+1:]}, \forall m. \tag{22}$$

3.5. Solution Algorithm

We propose a fast algorithm for LATC-Tubal as shown in Algorithm 1, which is well-suited to spatiotemporal data imputation problems. We can see that this algorithm does not have too many parameters. Here, ρ is the most basic parameter that controls the whole ADMM and the tensor singular value thresholding process. λ is a trade-off parameter between the autoregressive norm and the tensor nuclear norm. According to Eq. (18) and (19), it can be typically set as $\lambda = c \cdot \rho$ where if $c = 1$, then it implies that these two norms have the same importance in the result.

Algorithm 1: imputer($\boldsymbol{Y}, \rho, \lambda$)

Initialize $\boldsymbol{\mathcal{T}}$ as zeros and $\boldsymbol{A}$ as small random values. Set $\mathcal{P}_\Omega(\boldsymbol{Z}) = \mathcal{P}_\Omega(\boldsymbol{Y})$ and $\ell = 0$.
while *not converged* **do**
    Update $\boldsymbol{\mathcal{X}}^{\ell+1}$ by Eq. (15);
    Compute $\hat{\boldsymbol{X}} = \mathcal{Q}^{-1}(\boldsymbol{\mathcal{X}}^{\ell+1})$;
    Update $\boldsymbol{Z}_{[:h_d]}^{\ell+1}$ by Eq. (18);
    for $m = 1$ **to** M **do**
        Build $\boldsymbol{V}_m$ on $\hat{\boldsymbol{X}}$, and then update $\boldsymbol{z}_{m,[h_d+1:]}^{\ell+1}$ by Eq. (19);
    Transform observation information by letting $\mathcal{P}_\Omega(\boldsymbol{Z}^{\ell+1}) = \mathcal{P}_\Omega(\boldsymbol{Y})$;
    for $m = 1$ **to** M **do**
        Update $\boldsymbol{a}_m^{\ell+1}$ by Eq. (22);
    Update $\boldsymbol{\mathcal{T}}^{\ell+1}$ by Eq. (10);
    $\ell := \ell + 1$;
    $\rho = \min\{1.05 \times \rho, \rho_{\max}\}$;
return recovered matrix $\hat{\boldsymbol{X}}$.

In this algorithm, some hidden settings are also needed to discuss. We initialize unitary transform matrix $\boldsymbol{\Phi}$ by the left eigenvectors of $\boldsymbol{\mathcal{Z}}_{(3)}$. In the iterative process, we update $\boldsymbol{\Phi}$ by the left eigenvectors of $\boldsymbol{\mathcal{Z}}_{(3)} - \boldsymbol{\mathcal{T}}_{(3)}/\rho$ for every 10 iterations. To alleviate the computational burden of updating the variable $\boldsymbol{A}$, we manually set the sampling rates in Eq. (22) as 20% and 10% for $5000 < IJ \leq 10000$ and $IJ > 10000$, respectively.

4. Experiments

In this section, we evaluate the imputation performance of the LATC-Tubal model using two large-scale and high-dimensional traffic data sets collected from California PeMS system. We measure our model by both imputation accuracy and computational cost.

4.1. California PeMS Data Sets

To show the advantages of LATC-Tubal for handling with large-scale and high-dimensional traffic data, we particularly choose two publicly available data sets collected from California transportation system (i.e., PeMS) as our benchmark data sets[1]:

- **PeMS-4W data set**: This data set contains freeway traffic speed collected from 11160 traffic measurement sensors over 4 weeks (the first 4 weeks in the year of 2018) with a 5-minute time resolution (288 time intervals per day) in California, USA. It can be arranged in a matrix of size 11160×8064 or a tensor of size $11160 \times 288 \times 28$ according to the spatial and temporal dimensions. Note that this data set contains about 90 million observations.

- **PeMS-8W data set**: This data set contains freeway traffic speed collected from 11160 traffic measurement sensors over 8 weeks (the first 8 weeks in the year of 2018) with a 5-minute time resolution (288 time intervals per day) in California, USA. It can be arranged in a matrix of size 11160×16128 or a tensor of size $11160 \times 288 \times 56$ according to the spatial and temporal dimensions. Note that this data set contains about 180 million observations.

It is not difficult to see that these two data sets are both large-scale and high-dimensional. In what follows, we create two missing patterns, i.e., random missing (RM) and non-random missing (NM), which are same as our prior work (Chen and Sun, 2020). Then according to the mechanism of RM and NM patterns, we mask certain amount of observations as missing values (i.e., 30%, 70%) in both two data sets, and the remaining partial observations are input data for imputing these masked entries. To assess the imputation performance, we use the actual values of the masked missing entries as the ground truth to compute the matrices MAPE and RMSE:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100, \quad \text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \tag{23}$$

where $y_i, \hat{y}_i, i = 1, ..., n$ are actual values and estimated/imputed values, respectively.

4.2. Baseline Imputation Models

To conduct imputation experiments on the large-scale and high-dimensional traffic data sets, we actually do not have many competing models because most developed imputation algorithms are not suitable for such 'big' data. Thus, we compare the proposed LATC-Tubal model with the following baseline models:

- BPMF: Bayesian Probabilistic Matrix Factorization (Salakhutdinov and Mnih, 2008). This is a fully Bayesian model of the standard matrix factorization using Markov chain Monte Carlo (MCMC) algorithm.

- BGCP: Bayesian Gaussian CP decomposition (Chen et al., 2019). This is a fully Bayesian tensor factorization model using MCMC algorithm.

- HaLRTC: High-accuracy Low-Rank Tensor Completion (Liu et al., 2013). This is an LRTC model which uses weighted tensor nuclear norm.

- LRTC-TNN: Low-Rank Tensor Completion with Truncated Nuclear Norm minimization (Chen et al., 2020). This model is built on HaLRTC where the weighted tensor nuclear norm is replaced by multiple truncated nuclear norms with a truncation rate parameter. It has been shown to be superior to TRMF (Yu et al., 2016), BTMF (Sun and Chen, 2019), and other two baseline models we also compared in this article, i.e., BGCP (Chen et al., 2019) and HaLRTC (Liu et al., 2013).

- LATC-TNN: Low-Rank Autoregressive Tensor Completion with Truncated Nuclear Norm minimization (Chen and Sun, 2020). This model is built on LRTC with weighted tensor nuclear norm by integrating both autoregressive process and truncated nuclear norm.

- LATC-DCT: We consider the LATC model by integrating discrete cosine transform (DCT) as a linear transform. If we remove the autoregressive norm in LATC-DCT, i.e., $\lambda = 0$, this model is indeed tensor nuclear norm minimization with DCT (i.e., TNN-DCT) proposed by Lu et al. (2019). Note that DCT is a special case of linear unitary transform. When comparing LATC-DCT to the proposed LATC-Tubal model, it is possible to see the importance of unitary transform in LATC-Tubal.

To make fair comparison, we use the following convergence criteria for the LRTC/LATC models:

$$\frac{\|\hat{\boldsymbol{\mathcal{X}}}^{\ell+1} - \hat{\boldsymbol{\mathcal{X}}}^{\ell}\|_F^2}{\|\mathcal{P}_\Omega(\boldsymbol{\mathcal{Y}})\|_F^2} < \epsilon \quad \text{or} \quad \frac{\|\hat{\boldsymbol{X}}^{\ell+1} - \hat{\boldsymbol{X}}^{\ell}\|_F^2}{\|\mathcal{P}_\Omega(\boldsymbol{Y})\|_F^2} < \epsilon, \tag{24}$$

where $\hat{\boldsymbol{\mathcal{X}}}^{\ell+1}$ and $\hat{\boldsymbol{\mathcal{X}}}^{\ell}$ denote the recovered tensors at the $(\ell+1)$th iteration and ℓth iteration, respectively. Similarly, $\hat{\boldsymbol{X}}^{\ell+1}$ and $\hat{\boldsymbol{X}}^{\ell}$ denote the recovered tensors at the $(\ell+1)$th iteration and ℓth iteration, respectively. In the following experiments, we set $\epsilon = 1 \times 10^{-3}$.

For the BPMF model, we set the low ranks as 100 and 10 for RM and NM scenarios, respectively. For the BGCP model, we set the low rank as 10 for both RM and NM scenarios due to the very high computational cost underlying the larger low rank. The the total numbers of burn-in iterations and Gibbs iterations are 500 and 100, respectively. For the family of LATCs (i.e., LATC-TNN, LATC-DCT, LATC-Tubal), we set $\lambda = 5 \times \rho$ and $\lambda = \frac{1}{2} \times \rho$ for RM and NM scenarios, respectively. Time lags of these models are set as $\mathcal{H} = \{1, ..., 6, I-2, ..., I+3\}$. In particular, we discuss LATC-Tubal in the cases of $\lambda = 0$ and $\lambda \neq 0$ because putting LATC-Tubal ($\lambda = 0$) and LATC-Tubal ($\lambda \neq 0$) together could help understand the importance of autoregressive norm in the LATC framework. The Python implementation and more detailed parameter setting of the proposed LATC-Tubal model and its competing models are available at https://github.com/xinychen/tensor-learning.

4.3. Results

Table 1 shows the results of LATC-Tubal and its competing models for missing traffic data imputation tasks. For RM scenarios on both PeMS-4W and PeMS-8W data sets, the proposed LATC-Tubal model achieves high accuracy which is close to the best performance achieved by LRTC-TNN and LATC-TNN. For NM scenarios, LATC-Tubal is inferior to the LRTC-TNN and LATC-TNN due to the whole daily missingness in each time series. By comparing with HaLRTC, we can see that LATC-Tubal ($\lambda = 0$) performs consistently better in all missing scenarios. This clearly shows that tensor nuclear norm minimization with unitary transform is superior to weighted tensor nuclear norm minimization. The comparison results between LATC-DCT and LATC-Tubal indicate that data-driven unitary transform is more capable of characterizing day-to-day correlation than discrete cosine transform.

Table 1: Performance comparison (in MAPE/RMSE) for RM and NM data imputation tasks on California PeMS-4W and PeMS-8W traffic speed data. The size of PeMS-4W is $11160 \times (288 \times 28)$ (PeMS-8W: $11160 \times (288 \times 56)$) where 11160 is the number of sensors and 288 is the number of time points per day.

	PeMS-4W				PeMS-8W			
	30%, RM	70%, RM	30%, NM	70%, NM	30%, RM	70%, RM	30%, NM	70%, NM
BPMF	3.41/3.06	3.68/3.29	5.18/4.46	5.77/5.09	3.88/3.43	4.10/3.62	5.33/4.56	5.51/4.72
BGCP	5.06/4.32	5.05/4.32	5.45/4.62	8.05/10.35	5.45/4.60	5.43/4.59	5.52/4.67	5.72/4.84
HaLRTC	1.98/1.73	2.84/2.49	5.24/4.19	7.17/5.19	2.07/1.81	3.17/2.74	4.79/3.98	6.06/4.67
LRTC-TNN	1.67/**1.55**	**2.32/2.13**	4.62/3.94	**5.47**/4.71	1.81/1.66	2.57/2.29	**4.27/3.69**	**5.31**/4.52
LATC-TNN	**1.65**/1.58	2.34/2.22	**4.46/3.88**	5.56/**4.68**	**1.72**/1.64	2.50/2.35	4.28/3.75	**5.31/4.45**
LATC-DCT	1.69/1.58	2.41/2.23	6.22/4.94	7.35/5.60	**1.72/1.61**	2.45/**2.27**	5.69/4.69	6.57/5.18
LATC-Tubal ($\lambda = 0$)	1.71/1.61	2.46/2.26	5.59/4.52	6.59/5.07	1.74/1.63	2.51/2.30	5.28/4.42	5.97/4.77
LATC-Tubal ($\lambda \neq 0$)	1.69/1.59	2.40/2.23	5.59/4.52	6.60/5.07	**1.72/1.61**	**2.44/2.27**	5.28/4.42	5.98/4.77

In Table 1, we see that LRTC-TNN and LATC-TNN can produce comparable imputation accuracy. However, observing the running time of imputation models in Table 2, it is not difficult to conclude that both LRTC-TNN and LATC-TNN are not suitable for these large-scale imputation tasks due to their high computational cost. Table 2 clearly shows that LATC-Tubal with unitary transformed tensor nuclear norm minimization scheme is the most computationally efficient one by comparing to the competing models. Therefore, from both Table 1 and 2, the results suggest that the proposed LATC-Tubal model is an efficient solution to large-scale traffic data imputation while still maintaining the promising imputation performance close to the state-of-the-art models.

Table 2: Running time (in minute) of imputation models on California PeMS-4W and PeMS-8W traffic speed data. We also report the number of iteration when the algorithm is converged.

	PeMS-4W				PeMS-8W			
	30%, RM	70%, RM	30%, NM	70%, NM	30%, RM	70%, RM	30%, NM	70%, NM
BPMF	728.00	741.11	107.31	120.08	1165.15	1156.77	166.14	213.70
BGCP	174.24	204.48	174.26	212.79	346.48	403.9	347.49	414.21
HaLRTC	88.35 (18)	215.33 (37)	176.31 (34)	321.20 (61)	245.41 (18)	505.22 (38)	473.43 (35)	785.40 (59)
LRTC-TNN	354.82 (74)	322.66 (67)	468.25 (100)	465.34 (100)	877.37 (67)	807.00 (60)	1302.68 (100)	1307.53 (100)
LATC-TNN	238.94 (46)	234.44 (45)	143.23 (28)	159.33 (31)	578.12 (39)	638.84 (41)	383.94 (27)	439.00 (30)
LATC-DCT	35.15 (46)	48.21 (61)	35.47 (47)	44.23 (58)	71.69 (46)	117.59 (64)	86.16 (47)	94.86 (58)
LATC-Tubal ($\lambda = 0$)	**16.96** (48)	**29.10** (69)	**17.16** (48)	**28.66** (69)	**42.75** (48)	**74.77** (70)	**43.59** (48)	**70.75** (70)
LATC-Tubal ($\lambda \neq 0$)	35.96 (48)	52.73 (69)	36.29 (48)	52.72 (69)	84.48 (49)	132.17 (70)	82.95 (48)	135.69 (69)

We further discuss the imputation of the LATC-Tubal ($\lambda \neq 0$) by providing the heatmap of unitary transform matrices in Figure 4 and the visualization of time series in Figure 5. As shown in Figure 4(a), the initial unitary transform matrix produced by the partially observed tensor only harnesses the most basic daily patterns where the first few columns can help identify weekday and weekend. In the following iterative process, the patterns underlying unitary transform matrix become more complicated and informative (see Figure 4(b-c)). Figure 5 shows that LATC-Tubal can produce masked time series points accurately by learning from partial observations.

5. Conclusion and Future Directions

In this work, we develop a Low-Tubal-Rank Autoregressive Tensor Completion (LATC-Tubal) model for large-scale spatiotemporal imputation with superior efficiency. In this model, we use tensor nuclear norm minimization scheme to model the inherent low-rank structure of traffic data (third-order tensors of sensor $\times$ time of day $\times$ day). This third-order structure allows us to efficiently implement the nuclear norm minimization on the unitary transformed matrix for each day, while the day-to-day correlation can be preserved by the unitary transform matrix. Our experiments show that, the proposed LATC-Tubal model is very computationally efficient while still maintaining the imputation accuracy close to the earlier state-of-the-art models. Experiments also shows that the data-driven unitary transform matrix can harness effective traffic patterns.
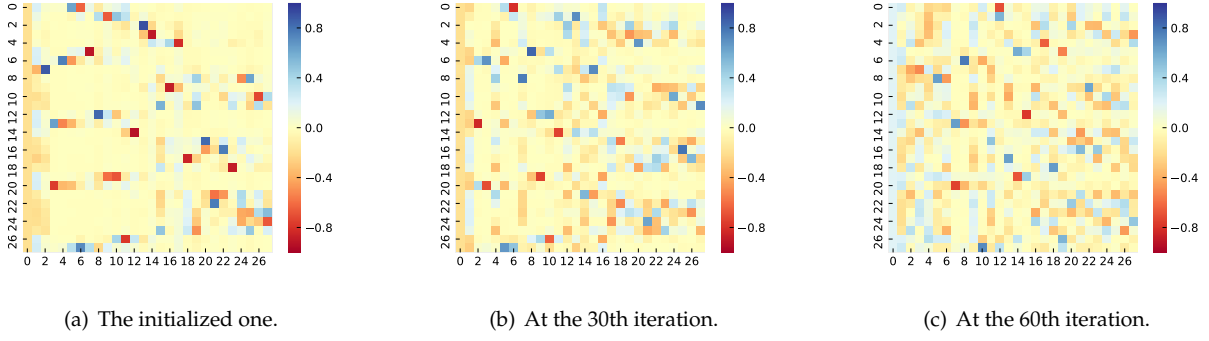
(a) The initialized one.
(b) At the 30th iteration.
(c) At the 60th iteration.

Figure 4: 28-by-28 unitary transform matrices of the PeMS-4W data set at the case of 70% RM scenario.

(a) The 1st time series.

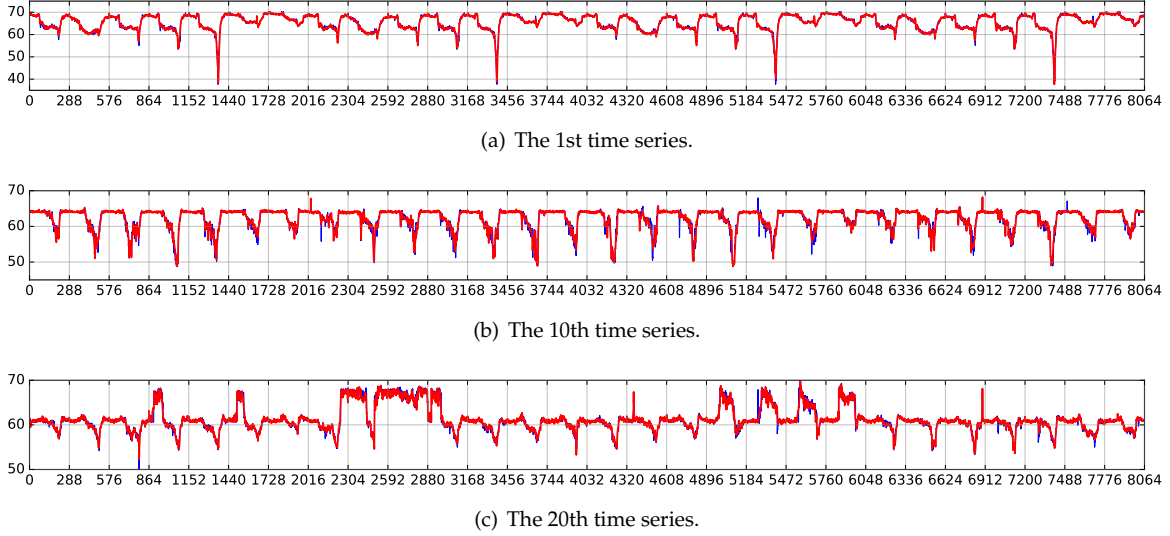(b) The 10th time series.

(c) The 20th time series.

Figure 5: Estimated time series (red curve) vs. ground truth (blue curve) for the PeMS-4W data set at the case of 70% RM scenario.

This paper also provides valuable insight into spatiotemporal data modeling, and there are also some future directions to advance this research:

- As shown in Lemma 1, the tensor singular value thresholding scheme is achieved by the singular value thresholding of matrix nuclear norm minimization. In the future, it would be feasible to adopt the truncated nuclear norm minimization for this scheme and improve the model performance.

- Following the LATC predictor proposed in Chen and Sun (2020), we can easily adapt the LATC-Tubal model to spatiotemporal traffic forecasting tasks. Unlike the costly retraining process of LATC predictor, LATC-Tubal would reinforce the LATC predictor and forecast traffic without retraining the models using a large amount of data.

Acknowledgement

Appendix

For the experimental evaluation, we used a platform consisting of a 3.5 GHz Intel Core i5-6600K processor (4 cores total) and 32 GB of RAM. It is also possible to run our code in a laptop with relatively smaller RAM. In the settings of software, we used Python 3.7 (with the Numpy and Pandas packages).

References

Asif, M. T., Mitrovic, N., Dauwels, J., Jaillet, P., 2016. Matrix and tensor based methods for missing data estimation in large traffic networks. IEEE Transactions on Intelligent Transportation Systems 17 (7), 1816–1825.

Asif, M. T., Mitrovic, N., Garg, L., Dauwels, J., Jaillet, P., 2013. Low-dimensional models for missing data imputation in road networks. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, pp. 3527–3531.

Bashir, F., Wei, H.-L., Feb. 2018. Handling missing data in multivariate time series using a vector autoregressive model-imputation (var-im) algorithm. Neurocomput. 276 (C), 2330.

Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y., apr 2018. Recurrent neural networks for multivariate time series with missing values. Scientific Reports 8 (1).

Chen, J., Shao, J., 2000. Nearest neighbor imputation for survey data. Journal of Official Statistics 16 (2), 113–132.

Chen, X., He, Z., Sun, L., 2019. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies 98, 73 – 84.

Chen, X., Sun, L., 2020. Low-rank autoregressive tensor completion for multivariate time series forecasting. arXiv 2006.10436.

Chen, X., Yang, J., Sun, L., 2020. A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies 117, 102673.

Kernfeld, E., Kilmer, M., Aeron, S., 2015. Tensortensor products with invertible linear transforms. Linear Algebra and its Applications 485, 545 – 570.

Kolda, T. G., Bader, B. W., 2009. Tensor decompositions and applications. SIAM Reviw 51 (3), 455–500.

Li, L., Su, X., Zhang, Y., Lin, Y., Li, Z., 2015. Trend modeling for traffic time series analysis: An integrated study. IEEE Transactions on Intelligent Transportation Systems 16 (6), 3430–3439.

Liu, J., Musialski, P., Wonka, P., Ye, J., 2013. Tensor completion for estimating missing values in visual data. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1), 208–220.

Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S., 2016. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).

Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S., 2020. Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (4), 925–938.

Lu, C., Peng, X., Wei, Y., June 2019. Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Qu, L., Li, L., Zhang, Y., Hu, J., 2009. Ppca-based missing data imputation for traffic flow volume: A systematical approach. IEEE Transactions on Intelligent Transportation Systems 10 (3), 512–522.

Qu, L., Zhang, Y., Hu, J., Jia, L., Li, L., June 2008. A BPCA based missing value imputing method for traffic flow volume data. In: 2008 IEEE Intelligent Vehicles Symposium. pp. 985–990.

Ran, B., Tan, H., Wu, Y., Jin, P. J., 2016. Tensor based missing traffic data completion with spatial-temporal correlation. Physica A: Statistical Mechanics and its Applications 446, 54–63.

Rao, K. R., Yip, P., 1990. Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press Professional, Inc., USA.

Salakhutdinov, R., Mnih, A., 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In: Proceedings of the 25th International Conference on Machine Learning. pp. 880–887.

Schafer, J. L., aug 1997. Analysis of Incomplete Multivariate Data. Chapman and Hall/CRC.

Song, G., Ng, M. K., Zhang, X., 2020. Robust tensor completion using transformed tensor singular value decomposition. Numerical Linear Algebra with Applications 27 (3), e2299.

Sun, L., Chen, X., 2019. Bayesian temporal factorization for multidimensional time series prediction. arXiv preprint arXiv:1910.06366.

Yu, H.-F., Rao, N., Dhillon, I. S., 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. In: Advances in neural information processing systems. pp. 847–855.