Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the manuscript number!)

Forecasting Urban Traffic States with Sparse Data Using Hankel Temporal Matrix Factorization

Xinyu Chen
Civil, Geological and Mining Engineering Department, Polytechnique Montréal, Montréal, QC H3T 1J4, Canada, chenxy346@gmail.com

Xi-Le Zhao
School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 610056, China, xlzhao122003@163.com

Chun Cheng*
School of Economics and Management, Dalian University of Technology, Dalian 116024, China,
*Corresponding author. chun.cheng@polymtl.ca

Forecasting urban traffic states is crucial to transportation network monitoring and management, playing an important role in the decision-making process. Despite the substantial progress that has been made in developing accurate, efficient, and reliable algorithms for traffic forecasting, most existing approaches fail to handle sparsity, high-dimensionality, and nonstationarity in traffic time series and seldom consider the temporal dependence between traffic states. To address these issues, this work presents a Hankel temporal matrix factorization (HTMF) model using the Hankel matrix in the lower-dimensional spaces under a matrix factorization framework. In particular, we consider an alternating minimization scheme to optimize the factor matrices in matrix factorization and the Hankel matrix in the lower-dimensional spaces simultaneously. To perform traffic state forecasting, we introduce two efficient estimation processes on real-time incremental data, including an online imputation (i.e., reconstruct missing values) and an online forecasting (i.e., estimate future data points). Through extensive experiments on the real-world Uber movement speed dataset in Seattle, USA, we empirically demonstrate the superior forecasting performance of HTMF over several baseline models and highlight the advantages of HTMF for addressing sparsity, nonstationarity, and short time series.

Key words: Urban transportation network, traffic state forecasting, machine learning, matrix factorization, Hankel matrix

History: August 12, 2024

1. Introduction

Traffic states of urban road networks reveal the efficiency of transportation systems, providing insights into urban transportation management. To enhance decision-making and system efficiency, it is essential to collect historical and real-time traffic states from urban road networks in a fine-grained spatiotemporal resolution and then produce reliable fore-

casts for the near-future traffic states. Recent transportation innovations have significantly simplified data collection processes, making real-world traffic data accessible. For instance, the development of sensing technologies enables the installation of location-based sensors on ridesharing vehicles to gather the traffic states of urban road networks (Furuhata et al. 2013). In particular, Uber has launched the Uber movement project to collect data on hourly traffic speeds from thousands of road segments (Hu et al. 2022).

To guarantee that high-quality decisions can be produced based on predictions, a sufficient number of samples is needed, meaning that a large number of ridesharing vehicles is expected for sensing city-wide traffic states (Treiber and Kesting 2013, Zheng 2015). However, challenges such as insufficient sampling of ridesharing vehicles in total traffic (Chen et al. 2022) and spatiotemporal sparsity of human mobility patterns (Liu et al. 2022) lead to the prevalent issue of missing data, being a substantial barrier to the prediction work. This phenomenon is formally introduced as sparsity, where only a small portion of the data is observed due to limitations in the data collection process (Qin et al. 2021, Chen et al. 2022). Another challenge is the high-dimensionality problem. High-dimensional traffic data often exhibit the characteristic that most relevant information is concentrated in only a few dimensions, necessitating dimensionality reduction algorithms.

Moreover, traffic states exhibit strong temporal dependence, with the current state being influenced by previous states and likely to impact future states. This dependence arises due to factors including traffic congestion, fluctuating travel demand, and the dynamic nature of traffic flow. For example, peak-hour congestion can trigger traffic cascades that persist throughout the day. Modeling temporal dependence is crucial for capturing the underlying dynamics and trends that shape urban traffic states. Thus, to extract valuable insights from future traffic forecasts, addressing the issues of sparsity, high-dimensionality, and temporal dependence is essential. However, these aspects, especially the temporal dependence between traffic states, have received limited attention, and the key challenges persist.

Matrix factorization, a data-driven machine learning approach, can effectively learn from sparse data and accurately reconstruct missing or unobserved values. It also shows potential in modeling spatiotemporal traffic states. However, matrix factorization models alone lack temporal correlations, unsuitable for forecasting future traffic states. On multivariate time series, integrating autoregressive processes into matrix factorization has revealed the

power of identifying temporal patterns from incomplete or even sparse time series data (Yu et al. 2016, Chen and Sun 2021, Chen et al. 2022). This is because autoregressive processes enable the establishment of temporal correlations on time series data. However, real-world time series are usually nonstationary, demonstrating clear changing levels, periodicity, or seasonality. Hence, it is unreasonable to directly build up an autoregressive process on latent temporal factors in matrix factorization (Chen et al. 2022), as the autoregressive processes are well-suited to the time series that are relatively stationary. Fortunately, alternative methodologies have been developed considering temporal dependence in the context of high-dimensional data without necessitating stationary patterns. For instance, Zhang et al. (2021) propose a dynamic recommender system to forecast grocery store sales. They employ non-parametric smoothing techniques and extrapolation methods on latent temporal factors to capture the temporal dynamics. In this work, we propose using a Hankel matrix within the matrix factorization framework to establish temporal correlations. This is because the Hankel matrix is flexible for automatically characterizing temporal correlations and dynamics using factorization or nuclear norm minimization. In contrast, autoregression should be well-designed in the matrix factorization considering complicated time series behavior. Moreover, the Hankel matrix shows great potential for short time series forecasting due to the principle of Hankel completion.

Hankel matrix arises from algebraic problems and has inspired considerable matrix/tensor completion algorithms (Cai et al. 2019, Sedighin et al. 2020). Its adoption in both machine learning and signal processing has many advantages: 1) it is an automatic delay embedding transformation (Yokota et al. 2018), 2) it enables an effective matrix completion and recovery (Cai et al. 2019), and 3) it demonstrates the capability of correcting corrupted columns in matrix completion tasks (Zhang and Wang 2019). Hankel structure has found diverse applications, such as visual data inpainting (e.g., missing slices) (Yokota et al. 2018) and time series forecasting (Shi et al. 2020, Sedighin et al. 2020). It is naturally combined with low-rank matrix completion (Cai et al. 2019, Zhang and Wang 2019). Nevertheless, Hankel matrix factorization/completion has its limitations for large-scale problems due to the explicit construction of large-scale Hankel structures.

In this work, to address the high-dimensionality issue, we utilize the Hankel structure and build up a Hankel matrix in the lower-dimensional latent spaces under a matrix factorization framework. We develop a novel Hankel temporal matrix factorization (HTMF) model,

which offers several advantages over existing methods such as those proposed in Yu et al. (2016), Chen and Sun (2021), and Chen et al. (2022). These existing methods correlate temporal factors through an autoregressive process in their temporal matrix factorization (TMF) models, whereas our HTMF model leverages a Hankel matrix on latent temporal factors to automatically achieve temporal modeling, which has the following benefits: 1) handling real-world nonstationary time series that demonstrate strong cyclical patterns and local time series trends; 2) estimating missing columns or even forecasting incoming columns in terms of spatiotemporal traffic states, analogous to the capability of existing TMF models; 3) presenting an efficient and economical representation on lower-dimensional latent temporal factors and demonstrating the capability of handling large-scale and high-dimensional data; 4) being suitable for short time series forecasting by using the latent Hankel matrix, instead of using Hankel structures on the data space as in Yokota et al. (2018), Sedighin et al. (2020), and Shi et al. (2020). To the best of our knowledge, most Hankel matrix-related algorithms are built on the data space, and HTMF is the first matrix factorization algorithm that establishes the Hankel matrix in the lower-dimensional spaces of matrix factorization.

Overall, our work makes the following contributions: 1) We propose an HTMF model for forecasting sparse, high-dimensional, and temporal-dependent traffic states. Hankel matrix built on latent temporal factors of matrix factorization allows an efficient estimation of temporal dynamics. To solve the model, we develop an alternating minimization scheme, including the conjugate gradient routine for estimating the spatial and temporal factor matrices and the hard thresholding process on the Hankel matrix. 2) We introduce both online imputation and online forecasting methods for traffic state estimation with real-time incremental data. The online imputation method reconstructs missing values in the incremental data, while the online forecasting method forecasts near-future traffic states. 3) We test the performance of HTMF using the Seattle Uber movement speed dataset, which is high-dimensional and sparse. Results show that HTMF outperforms several baseline models in handling nonstationarity, sparsity, and short time series.

In the rest of this paper, Section 2 reviews related studies. Section 3 introduces the HTMF model based on matrix factorization and Hankel matrix. For traffic state forecasting, we propose both online imputation and forecasting methods for real-time incremental traffic state data. Section 4 reports numerical results. Section 5 concludes the paper.

2. Literature Review

Substantial progress has been made in the past decades in developing computationally efficient machine learning algorithms to forecast spatiotemporal traffic. Despite the considerable growth and richness of the existing literature on traffic forecasting, a compelling question persists: How can high-dimensional and sparse traffic time series be forecasted?

2.1. Time Series Forecasting with Missing Values

Recently, various time series forecasting approaches have been developed, including but not limited to the classical statistical methods such as autoregressive models (Hamilton 1994) and deep learning methods (Lim and Zohren 2021). Nevertheless, most of these are not well-suited to forecasting incomplete time series data, as the phenomenon of missing data makes the forecasting task more complicated. Due to the difficulty of modeling sparse time series, past efforts devoted to this topic are limited. Anava et al. (2015) provide an autoregressive predictor for generating the time series from incomplete data. While some preliminary research on deep learning finds a solution that first imputes missing values and then forecasts time series on the reconstructed data (Che et al. 2018), Tang et al. (2020) propose to jointly model local and global temporal dynamics of multivariate time series from partial observations. These methods do not satisfy the modeling needs for handling complicated missing data patterns (Chen and Sun 2021) and high-dimensionality (Sen et al. 2019). In some cases, the high dimensionality could complicate the time series forecasting with missing values, posing methodological challenges.

Complementary to the above methods, another direction stimulated by autoregression and matrix/tensor factorization models shows great potential for modeling sparse time series. Xiong et al. (2010) demonstrate the importance of temporal dynamics when modeling collaborative filtering with tensor factorization and temporal smoothing. Following that idea, Yu et al. (2016) present a temporal regularized matrix factorization by integrating a univariate autoregressive process into a matrix factorization framework, and Gultekin and Paisley (2018) introduce an online matrix factorization that can learn low-dimensional embeddings and produce recursive estimates through an autoregressive process. Such models address the sparsity issue and are well-suited to high-dimensional time series. Beyond these models, Chen and Sun (2021) propose a Bayesian temporal matrix/tensor factorization framework by utilizing the same principle, and the core of temporal modeling is the

vector autoregression (VAR) process. However, since the stationarity requirement of real-world time series does not always hold, these methods still cannot fit the practical needs. To this end, Chen et al. (2022) reinforce these methods and address the nonstationarity issue by considering the differencing operations (e.g., seasonal differencing) when using autoregressive processes in matrix factorization. The authors propose the TMF model that tackles both high-dimensionality and sparsity; however, differencing operations such as first-order differencing and seasonal differencing are not flexible enough to eliminate the nonstationarity of time series.

2.2. Spatiotemporal Traffic Imputation and Forecasting

Spatiotemporal traffic data are usually incomplete due to spatial and temporal sparsity, e.g., the sparsity characteristics of Uber movement data. Many approaches have been proposed for imputing missing or unobserved traffic data in the spatiotemporal context. Examples include principle component analysis (Li et al. 2013), multi-output Gaussian process (Rodrigues et al. 2018), Bayesian tensor factorization (Chen et al. 2019), low-rank tensor completion (Chen et al. 2020), Bayesian TMF (Chen and Sun 2021), etc. Beyond the imputation task, TMF models with VAR in the lower-dimensional latent spaces are illustrated to be effective for the spatiotemporal traffic forecasting on sparse data (Chen and Sun 2021, Chen et al. 2022). TMF models bridge the gap between imputation and forecasting with missing values because they can learn low-dimensional spatiotemporal patterns and perform forecasting without imputation.

3. Problem Definition and Methodology

This section introduces the problem and develops an HTMF model for spatiotemporal traffic state forecasting. The HTMF model consists of a spatial factor matrix, a temporal factor matrix, and a Hankel matrix. In particular, HTMF establishes a Hankel matrix on the temporal factor matrix for exhibiting lower-dimensional temporal patterns. To adopt the proposed model to traffic state forecasting tasks, we finally elaborate on the forecasting mechanism of HTMF, including both online imputation and online forecasting for real-time incremental traffic state data.

3.1. Problem Definition

Let $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ with columns $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T \in \mathbb{R}^N$ represent the traffic state measurements collected from N road segments with T time steps (i.e., time intervals with a certain

resolution), and Ω denote the index set of observed entries in $\boldsymbol{Y}$, then the goal is to forecast $\boldsymbol{y}_{T+1}, \ldots, \boldsymbol{y}_{T+\delta}, \delta \in \mathbb{N}^+$ (δ is the forecasting time horizon). Figure 1 illustrates the forecasting task.

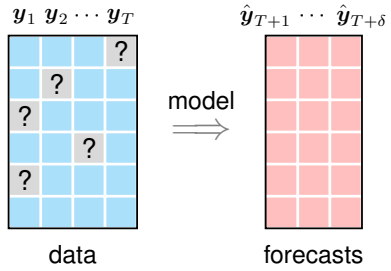


Figure 1 **Illustration of the traffic state forecasting task on the incomplete data $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T$.**

For traffic state forecasting, we aim at simultaneously handling the following characteristics of traffic state data: 1) *High-dimensionality*: data is of high dimension and large scale with thousands of road segments. 2) *Sparsity*: only a small fraction of data are observed. To obtain satisfactory forecasting, we must address the following concerns: 1) How to perform traffic state forecasting with a small fraction of observations? 2) How to apply the TMF framework to a large-scale and high-dimensional setting?

Unlike the classical traffic forecasting approaches, which require fully observed data as inputs, our work addresses traffic state forecasting on imperfect and sparse data. In the following, we consider a data-driven approach by applying the low-rank matrix factorization framework, expecting to simultaneously address the high-dimensionality and sparsity issues in traffic state data.

3.2. Model Description

The collection of extensive traffic state data from urban road networks has necessitated the exploration of data-driven methodologies for practical applications and prompted a reevaluation of strategies to effectively address the challenges posed by high-dimensionality and sparsity (Chen and Sun 2021). Given $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$, the objective is to reconstruct missing entries in $\boldsymbol{Y}$ with partial observations. Let $\mathcal{P}_\Omega(\boldsymbol{Y})$ denote the orthogonal projection supported on the observed index set Ω, defined as

$$[\mathcal{P}_\Omega(\boldsymbol{Y})]_{it} = \begin{cases} y_{it}, \text{ if } (i,t) \in \Omega, \\ 0, \quad \text{otherwise}, \end{cases} \tag{1}$$

where $i = 1, \ldots, N$ and $t = 1, \ldots, T$. Notably, $[\cdot]_{it}$ denotes the (i,t)-th entry of a matrix. Low-rank matrix factorization is one of the most classical approaches for achieving the reconstruction objective (Mnih and Salakhutdinov 2007, Koren et al. 2009), taking the form of

$$\min_{\boldsymbol{W},\boldsymbol{X}} \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})\right\|_F^2+\frac{\rho}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2), \tag{2}$$

where $R < \min\{N, T\}$ is the low rank of this matrix factorization. $\boldsymbol{W} \in \mathbb{R}^{R\times N}$ and $\boldsymbol{X} \in \mathbb{R}^{R\times T}$ are spatial and temporal factor matrices, respectively. The symbol $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. ρ is the trade-off parameter for regularization.

However, the standard matrix factorization cannot incorporate local temporal correlations into global low-rank patterns. To this end, Yu et al. (2016), Chen and Sun (2021), and Chen et al. (2022) develop some TMF models that apply the dth-order VAR process, which can be summarized as the following optimization problem:

$$\min_{\boldsymbol{W},\boldsymbol{X},\{\boldsymbol{A}\}_{k=1}^{d}} \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})\right\|_F^2+\frac{\rho}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2)+\frac{\lambda}{2}\sum_{t=d+1}^{T}\left\|\boldsymbol{x}_t-\sum_{k=1}^{d}\boldsymbol{A}_k\boldsymbol{x}_{t-k}\right\|_2^2, \tag{3}$$

where the vector $\boldsymbol{x}_t, \forall t \in \{1, \ldots, T\}$ is the t-th column (time snapshot) of $\boldsymbol{X}$, and $\boldsymbol{A}_k \in \mathbb{R}^{R\times R}, k = 1, \ldots, d$ are the coefficient matrices of VAR on temporal factors. The symbol $\|\cdot\|_2$ denotes the ℓ_2-norm of a vector. ρ and λ are the trade-off parameters for regularization. Since VAR is important in temporal modeling, TMF allows one to make accurate missing data imputations and effective high-dimensional time series forecasting (Yu et al. 2016, Chen and Sun 2021). However, one dominant assumption—stationarity of time series—poses a bottleneck for the broad use of TMF.

In this work, we intend to seek an effective temporal modeling technique and take advantage of the Hankel matrix in a TMF framework to address the nonstationarity of traffic states. In particular, we construct a Hankel matrix on the temporal factor matrix $\boldsymbol{X}$ utilizing Definition 1 (Zhang et al. 2018, Zhang and Wang 2019).

DEFINITION 1 (HANKEL MATRIX). Given a multivariate time series data matrix $\boldsymbol{X} \in \mathbb{R}^{R\times T}$ with columns $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T \in \mathbb{R}^R$ and a window length $d \in \mathbb{N}^+$, let $\mathcal{H}_d : \mathbb{R}^{R\times T} \to \mathbb{R}^{(dR)\times m}$ be a linear operator which maps $\boldsymbol{X}$ into the following Hankel matrix:

$$\mathcal{H}_d(\boldsymbol{X}) = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_m \\ \boldsymbol{x}_2 & \boldsymbol{x}_3 & \cdots & \boldsymbol{x}_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{x}_d & \boldsymbol{x}_{d+1} & \cdots & \boldsymbol{x}_T \end{bmatrix} \in \mathbb{R}^{(dR)\times m}, \tag{4}$$

with $m = T - d + 1$.

As is well known in the time series analysis, VAR usually assumes the time series to be stationary. Therefore, Hankel representation produces temporal correlations that VAR is unable to produce efficiently, and it is reasonable to postulate that the Hankel matrix can reinforce temporal modeling. In this work, our HTMF model follows the form:

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{X}} \ & \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})\right\|_F^2+\frac{\rho}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2) \\ \text{s.t. } & \operatorname{rank}(\mathcal{H}_d(\boldsymbol{X}))=R, \end{aligned} \tag{5}$$

where $\operatorname{rank}(\cdot)$ denotes the rank of a matrix. $\mathcal{H}_d(\boldsymbol{X})$ is assumed to be rank-R with $R \leq T-d-1$. This implies that the $(dR) \times m$ Hankel matrix on lower-dimensional latent temporal factors is an efficient and economical representation. Alternatively, if one builds a Hankel matrix on the original traffic state data $\boldsymbol{Y}$, there would be a Hankel matrix of size $(dN) \times m$, possibly consuming high memory usage and computational cost. Moreover, the HTMF model captures temporal correlations within d window length through the Hankel structure (i.e., the same blocks on the anti-diagonal), and the window length d is usually smaller than the number of time steps per day or week; thus, the Hankel structure in HTMF only captures local temporal correlations, which makes it possible for HTMF to handle real-world nonstationary time series. Figure 2 illustrates the HTMF, whose ingredients include a spatial factor matrix, a temporal factor matrix, and a Hankel matrix.

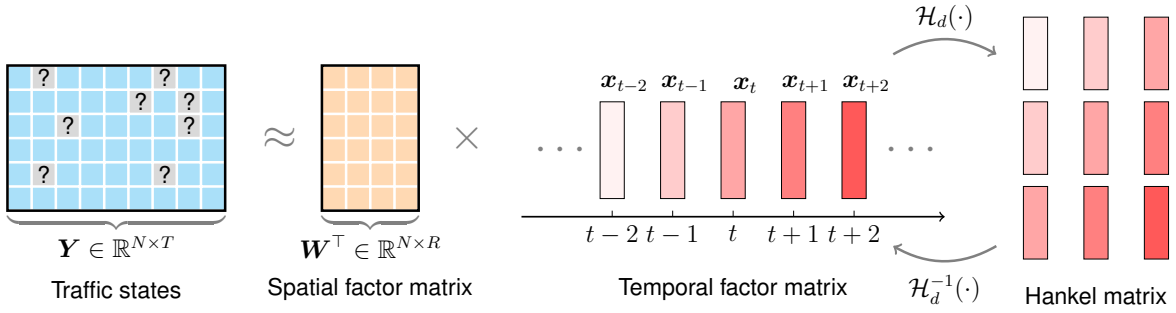


Figure 2 **Illustration of HTMF for modeling sparse traffic state data.**

The basic idea of HTMF is closely related to the low-rank Hankel matrix completion proposed by Cai et al. (2019). Matrix completion refers to recovering a matrix from its partially revealed entries. With no further assumptions, matrix completion is an ill-posed problem that does not even have a unique solution. Thus, to develop efficient solution

algorithms, researchers often have to exploit the inherent simplicity of the target matrix. A prominent example is the low-rank matrix completion, where the target matrix is assumed to be low-rank (Chen et al. 2021). Based on the property of low-rankness, various efficient algorithms are designed (Chen and Chi 2018). Going back to the HTMF, the constraint in the optimization problem (5) is expected to achieve temporal modeling through the low-rankness of the Hankel matrix. To solve the optimization problem in (5), we introduce an auxiliary variable $\boldsymbol{F} \in \mathbb{R}^{R\times T}$ to guarantee the low-rankness of the Hankel matrix. The model now becomes

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{X},\boldsymbol{F}} \ & \frac{1}{2}\left\|\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})\right\|_F^2+\frac{\rho}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2)+\frac{\gamma}{2}\|\boldsymbol{F}-\boldsymbol{X}\|_F^2 \\ \text{s.t. } & \operatorname{rank}(\mathcal{H}_d(\boldsymbol{F}))=R, \end{aligned} \tag{6}$$

where γ is the trade-off parameter for the additional regularization.

Regarding this model, we consider an alternating minimization scheme. Let f denote the objective function of problem (6), then our scheme can be basically summarized as follows,

$$\begin{cases} \boldsymbol{W}_{\ell+1}=\{\boldsymbol{W} \mid \frac{\partial f}{\partial \boldsymbol{W}}=\boldsymbol{0}\}, \\ \boldsymbol{X}_{\ell+1}=\{\boldsymbol{X} \mid \frac{\partial f}{\partial \boldsymbol{X}}=\boldsymbol{0}\}, \\ \boldsymbol{F}_{\ell+1}=\underset{\operatorname{rank}(\mathcal{H}_d(\boldsymbol{F}))=R}{\arg\min} \frac{1}{2}\|\boldsymbol{F}-\boldsymbol{X}_{\ell+1}\|_F^2, \end{cases} \tag{7}$$

where ℓ is the iteration counter.

In our original optimization problem, we have two variables, i.e., spatial factor matrix $\boldsymbol{W}$ and temporal factor matrix $\boldsymbol{X}$. However, the variable $\boldsymbol{X}$ is associated with both partially observed matrix factorization and low-rank Hankel matrix constraints. Thus, finding a closed-form solution for $\boldsymbol{X}$ becomes challenging. Problem (6) presents a readily implementable formula that incorporates a weight parameter γ to guarantee that $\boldsymbol{X}$ and $\boldsymbol{F}$ are as close as possible. If $\boldsymbol{F}$ is a numerical approximation of $\boldsymbol{X}$, depending on γ, then problem (6) is consistent with problem (5).

The partial derivative of f with respect to $\boldsymbol{W}$ can be written as

$$\frac{\partial f}{\partial \boldsymbol{W}}=-\boldsymbol{X}\mathcal{P}_{\Omega}^{\top}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})+\rho\boldsymbol{W}, \tag{8}$$

where $\frac{\partial f}{\partial \boldsymbol{W}}=\boldsymbol{0}$ admits a closed-form least-squares solution with respect to $\boldsymbol{W}$ (Koren et al. 2009). Alternatively, we can use the conjugate gradient method to obtain an approximated numerical solution to $\boldsymbol{W}$ from the generalized Sylvester equation (Chen et al. 2022):

$$\boldsymbol{X}\mathcal{P}_{\Omega}^{\top}(\boldsymbol{W}^{\top}\boldsymbol{X})+\rho\boldsymbol{W}=\boldsymbol{X}\mathcal{P}^{\top}(\boldsymbol{Y}). \tag{9}$$

If one uses the conjugate gradient method to find an approximated solution to the above equation, it should convert the matrix equation into a standard system of linear equations. First, we introduce an operator to represent the left-hand side of the equation as follows,

$$\mathcal{L}_w(\boldsymbol{W}) \triangleq \text{vec}(\boldsymbol{X}\mathcal{P}_\Omega^\top(\boldsymbol{W}^\top\boldsymbol{X}) + \rho\boldsymbol{W}), \tag{10}$$

where $\text{vec}(\cdot)$ denotes the vectorization operator. Then, with this definition in place, we apply the conjugate gradient method (Golub and Van Loan 2013) as an efficient matrix equation solver. Since (10) is a standard system of linear equations whose matrix is real symmetric and positive-definite, it satisfies the criteria of the conjugate gradient method. As revealed by Rao et al. (2015), the conjugate gradient-based alternating minimization scheme for matrix factorization is efficient and scalable to large applications. Notably, the conjugate gradient method is known to search for the solution to a system of linear equations with a relatively small number of iterations (e.g., 5 or 10). We summarize the algorithm of the conjugate gradient for estimating spatial factor matrix $\boldsymbol{W}$ in Algorithm 1.

Algorithm 1 Conjugate gradient for estimating spatial factor matrix $\boldsymbol{W}$

Input: Data matrix $\boldsymbol{Y}$, Ω as the observed index set, initialized spatial factor matrix $\boldsymbol{W}$, temporal factor matrix $\boldsymbol{X}$, and the maximum iteration n_w.

Output: Estimated spatial factor matrix $\boldsymbol{W}$.

Initialize $\boldsymbol{w}_0$ by the vectorized $\boldsymbol{W}$.
Compute residual vector $\boldsymbol{r}_0 = \text{vec}(\boldsymbol{X}\mathcal{P}_\Omega^\top(\boldsymbol{Y})) - \mathcal{L}_w(\boldsymbol{W})$, and let $\boldsymbol{q}_0 = \boldsymbol{r}_0$.
for $\ell = 0$ to $n_w - 1$ **do**
    Convert vector $\boldsymbol{q}_\ell$ into matrix $\boldsymbol{Q}_\ell$.
    Compute $\alpha_\ell = \frac{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}{\boldsymbol{q}_\ell^\top \mathcal{L}_w(\boldsymbol{Q}_\ell)}$.
    Update $\boldsymbol{w}_{\ell+1} = \boldsymbol{w}_\ell + \alpha_\ell \boldsymbol{q}_\ell$.
    Update $\boldsymbol{r}_{\ell+1} = \boldsymbol{r}_\ell - \alpha_\ell \mathcal{L}_w(\boldsymbol{Q}_\ell)$.
    Compute $\beta_\ell = \frac{\boldsymbol{r}_{\ell+1}^\top \boldsymbol{r}_{\ell+1}}{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}$.
    Update $\boldsymbol{q}_{\ell+1} = \boldsymbol{r}_{\ell+1} + \beta_\ell \boldsymbol{q}_\ell$.
end for
Convert vector $\boldsymbol{w}_{n_w}$ into matrix $\boldsymbol{W}$.
return Spatial factor matrix $\boldsymbol{W}$.

With respect to the temporal factor matrix $\boldsymbol{X}$, we can write down the partial derivative of f and obtain the solution to $\boldsymbol{X}$ from the following matrix equation:

$$\frac{\partial f}{\partial \boldsymbol{X}} = -\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{Y} - \boldsymbol{W}^{\top}\boldsymbol{X}) + \rho\boldsymbol{X} + \gamma(\boldsymbol{X} - \boldsymbol{F}) = \boldsymbol{0}, \tag{11}$$

or equivalently,

$$\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{W}^{\top}\boldsymbol{X}) + (\rho + \gamma)\boldsymbol{X} = \boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{Y}) + \gamma\boldsymbol{F}. \tag{12}$$

To solve this matrix equation with the conjugate gradient method, we can also define an operator that represents the vectorization of the left-hand part of the equation:

$$\mathcal{L}_{x}(\boldsymbol{X}) = \mathrm{vec}(\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{W}^{\top}\boldsymbol{X}) + (\rho + \gamma)\boldsymbol{X}), \tag{13}$$

then we follow a similar routine as shown in Algorithm 1 for estimating the approximated solution to $\boldsymbol{X}$.

In HTMF, rank-R Hankel matrix $\mathcal{H}_d(\boldsymbol{F})$ in the constraint allows one to build temporal correlations implicitly, and it follows a hard singular value thresholding (HSVT, see Definition 2) process (Cai et al. 2019).

Definition 2 (HSVT). For any $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, the hard thresholding operator $\mathcal{T}_r(\cdot)$ with positive integer $r < \min\{m, n\}$ is defined as

$$\mathcal{T}_r(\boldsymbol{X}) = \sum_{k=1}^{r} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^{\top}, \tag{14}$$

where $\boldsymbol{X} = \sum_{k=1}^{\min\{m,n\}} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^{\top}$ is the singular value decomposition (SVD) of $\boldsymbol{X}$ and its singular values are sorted as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}}$.

Definition 3 (Inverse Hankel matrix). Following Definition 1, given any matrix $\boldsymbol{Z} \in \mathbb{R}^{(dR)\times m}, m = T - d + 1$, the (i, j)-th entry of $\mathcal{H}_d^{-1}(\boldsymbol{Z}) \in \mathbb{R}^{R\times T}$ is defined as follows,

$$[\mathcal{H}_d^{-1}(\boldsymbol{Z})]_{ij} = \frac{1}{w_j} \sum_{k_1 + k_2 = j+1} z_{(k_1-1)R+i, k_2}, \tag{15}$$

where w_j denotes the number of elements in the j-th anti-diagonal of a $d \times m$ matrix.

Let the SVD of $\mathcal{H}_d(\boldsymbol{X})$ be

$$\mathcal{H}_d(\boldsymbol{X}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} = \sum_{r=1}^{\min\{dR,m\}} \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^{\top}, \text{ such that } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{dR,m\}}, \tag{16}$$

then the HSVT with any positive integer $R < \min\{dR, m\}$ can be written as follows,

$$\mathcal{T}_R(\mathcal{H}_d(\boldsymbol{X})) = \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^\top. \tag{17}$$

In the iterative process, the result of (17) is used to update the temporal factor matrix:

$$\boldsymbol{F} := \mathcal{H}_d^{-1}(\mathcal{T}_R(\mathcal{H}_d(\boldsymbol{X}))), \tag{18}$$

where $\mathcal{H}_d^{-1} : \mathbb{R}^{(dR)\times m} \to \mathbb{R}^{R\times T}$ is the inverse operator of $\mathcal{H}_d : \mathbb{R}^{R\times T} \to \mathbb{R}^{(dR)\times m}$. Since $\mathcal{H}_d$ is injective, $\mathcal{H}_d^{-1}$ follows from Zhang and Wang (2019) that the entry of inverse Hankel matrix is the average of certain anti-diagonal entries in the Hankel matrix (see Definition 3). Once the result of HSVT is available, the inverse operator can convert a given Hankel matrix into the expected form.

We illustrate the alternating minimization scheme for the HTMF model in Figure 3. For the spatial factor matrix $\boldsymbol{W}$ and the temporal factor matrix $\boldsymbol{X}$, we use the conjugate gradient method to approximate the least-squares solution. To fulfill the temporal modeling on temporal factors, we first implement HSVT on the Hankel matrix and then update the temporal factor matrix by the result of HSVT. The detailed algorithm of HTMF is summarized in Algorithm 2.

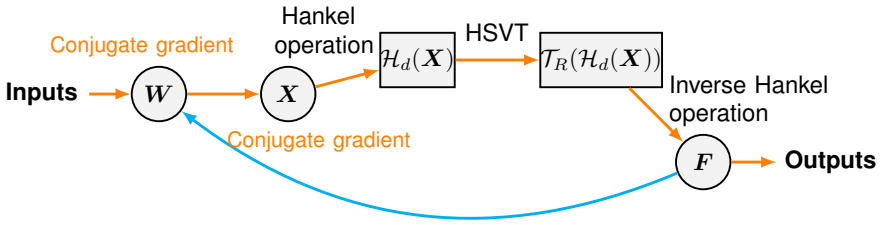


Figure 3 **Illustration of the alternating minimization scheme for implementing HTMF. Starting from the input data and initial parameters, we learn an HTMF model by updating the involved variables iteratively.**

Figure 4 presents the empirical time complexity of HTMF (rank $R = 10$) under various data lengths. The synthetic data is generated randomly with the size $NT = 1 \times 10^7, 2 \times 10^7, \ldots, 10 \times 10^7$. The window length of HTMF is set as $d = 10$ and 20, respectively. The results indicate that the time complexity exhibits a near-linear pattern in relation to the data size. This finding implies that the computing time of HTMF is expected to increase linearly as the dataset size expands, thereby demonstrating the predictability of computing time with respect to data volume.

Algorithm 2 HTMF($\boldsymbol{Y}, \Omega, d, R, n$)

Input: Data matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$, Ω as the observed index set, d as the window length, R as the low rank ($R \leq \min\{N, T-d-1\}$), and n as the maximum iteration.

Output: Spatial factor matrix $\boldsymbol{W}$ and temporal factor matrix $\boldsymbol{X}$.

1: Initialize $\boldsymbol{W}, \boldsymbol{X}$.
2: **for** $\ell = 0$ to $n-1$ **do**
3: Compute $\boldsymbol{W}$ from (9) with conjugate gradient.
4: Compute $\boldsymbol{X}$ from (11) with conjugate gradient.
5: Implement HSVT on $\mathcal{H}_d(\boldsymbol{X})$ by (17).
6: Update $\boldsymbol{F}$ by (18).
7: **end for**

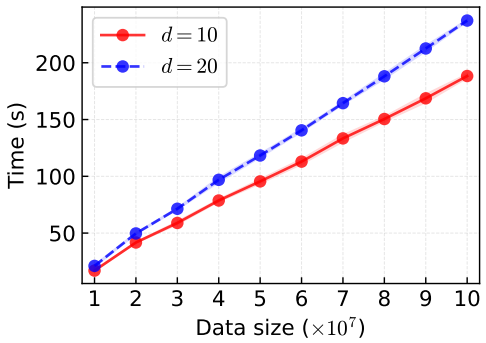


Figure 4 **Empirical time complexity of HTMF on the synthetic data. The HTMF model is tested 10 times on each randomly generated data, and average results are reported. Note that the data size is NT where $N = 10^3$ and $T \in \{1 \times 10^4, 2 \times 10^4, \ldots, 10 \times 10^4\}$.**

3.3. Forecasting Algorithm

To cater for real-world traffic state forecasting needs, we focus on a rolling forecasting task as the mechanism illustrated in Figure 5. As can be seen, the incremental input data are $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_3\}, \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_5\}, \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_7\}$, corresponding to the first, the second, and the third rolling times, respectively. At each rolling time, the forecasting time horizon is set to $\delta = 2$ and therefore we can finally gather $\{\hat{\boldsymbol{y}}_4, \ldots, \hat{\boldsymbol{y}}_9\}$ as forecasts in the rolling forecasting task.

Since the incremental input data have a certain number of missing values, the estimation tasks include both online imputation and online forecasting processes. Regarding the imputation process, the online imputation model is required to reconstruct missing values from the real-time incremental observations efficiently. To carry out an efficient estimation,

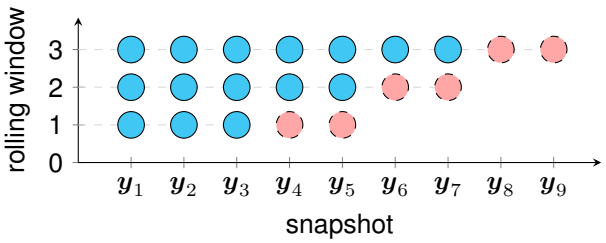


Figure 5 **Graphical scheme of the rolling forecasting on traffic states. The snapshot $\boldsymbol{y}_t \in \mathbb{R}^N, \forall t \in \{1, \ldots, T\}$ refers to the t-th column of $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$.**

we assume that the online imputation admits a matrix factorization, formulated as

$$\min_{\boldsymbol{X}} \frac{1}{2}\|\mathcal{P}_{\tilde{\Omega}}(\tilde{\boldsymbol{Y}} - \boldsymbol{W}^\top \boldsymbol{X})\|_F^2 + \frac{\rho}{2}\|\boldsymbol{X}\|_F^2, \tag{19}$$

where $\tilde{\boldsymbol{Y}}$ is the incremental traffic state data and $\tilde{\Omega}$ is the corresponding observed index set. The fixed variable $\boldsymbol{W}$ is estimated from the original traffic state data. Clearly, this optimization problem is convex and thus $\boldsymbol{X}$ has an optimal solution. Let g denote the objective of this optimization problem, then we have the following matrix equation:

$$\frac{\partial g}{\partial \boldsymbol{X}} = -\boldsymbol{W}\mathcal{P}_{\tilde{\Omega}}(\tilde{\boldsymbol{Y}} - \boldsymbol{W}^\top \boldsymbol{X}) + \rho \boldsymbol{X} = \boldsymbol{0}. \tag{20}$$

This is similar to the above problem associated with the temporal factor matrix. In this matrix equation, we estimate variable $\boldsymbol{X}$ through the conjugate gradient method. Accordingly, the missing entries in $\tilde{\boldsymbol{Y}}$ can be reconstructed by $\boldsymbol{W}^\top \boldsymbol{X}$.

The online forecasting in this work has two procedures. We first build an augmented temporal factor matrix as $\tilde{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{0}_{R \times \delta} \end{bmatrix}$ in which the last δ columns are filled with zeros (i.e., corrupted columns). In a Hankel matrix representation, the estimation of the last δ unknown columns is equivalent to the reconstruction of $\mathcal{H}_d(\tilde{\boldsymbol{X}})$ from partially revealed entries. Therefore, the online forecasting process admits a Hankel matrix factorization, which is formulated as follows,

$$\min_{\boldsymbol{V}} \frac{1}{2}\|\mathcal{P}_{\Omega_x}(\mathcal{H}_d(\tilde{\boldsymbol{X}}) - \boldsymbol{U}\boldsymbol{V})\|_F^2, \tag{21}$$

where Ω_x stands for the observed index set of the Hankel matrix $\mathcal{H}_d(\tilde{\boldsymbol{X}})$. In particular, we use $\boldsymbol{U} \in \mathbb{R}^{(dR) \times R}$ that consists of the left singular vectors $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_R\}$ of $\mathcal{H}_d(\boldsymbol{X})$ as the fixed variable (obtained in Algorithm 2). The solution to (21) can be derived from the following matrix equation:

$$\boldsymbol{U}^\top \mathcal{P}_{\Omega_x}(\boldsymbol{U}\boldsymbol{V}) = \boldsymbol{U}^\top \mathcal{P}_{\Omega_x}(\mathcal{H}_d(\tilde{\boldsymbol{X}})). \tag{22}$$

Let

$$\mathcal{L}_v(\boldsymbol{V}) = \text{vec}(\boldsymbol{U}^\top \mathcal{P}_{\Omega_x}(\boldsymbol{U}\boldsymbol{V})), \tag{23}$$

be an operator associated with variable $\boldsymbol{V}$, then we can estimate the solution to $\boldsymbol{V}$ through the conjugate gradient method. In this case, together with the inverse Hankel operation, estimating $\boldsymbol{V}$ allows one to produce the forecasts of the last δ columns in $\tilde{\boldsymbol{X}}$. Thus, the Hankel matrix is an effective representation for automatically filling in the last δ columns of the temporal factor matrix $\tilde{\boldsymbol{X}}$, due to its inherent anti-diagonal structure. Both online imputation and online forecasting processes do not consume too much computing resources. The convex problems can be solved efficiently in the rolling forecasting task.

Finally, we emphasize that online imputation is unnecessary for the online forecasting task. However, in the rolling forecasting scheme, it becomes essential to capture the temporal factors from the input incremental data that contains missing values. This process enables us to obtain low-dimensional latent temporal factors represented as a sequence of time series. By leveraging the Hankel structure, we can directly forecast future latent temporal factors and subsequently employ the multiplication of spatial and temporal factors to generate traffic state forecasts.

4. Experiments on Uber Movement Speed Data

In this section, we use the Seattle Uber movement speed dataset to evaluate our method. The nature of traffic state measurements, like the Uber movement data capturing hourly speed per road segment, often exhibits a pronounced indication of potential periodic and seasonal temporal patterns due to different mechanisms. These mechanisms encompass distinct time ranges as well as weather and seasonality. Each mechanism operates on different time scales within periodic cycles, with some exhibiting long-term patterns and others displaying short-term variations. However, matrix factorization serves as a similarity-based model. For instance, it can leverage users' latent factors in recommender systems to estimate the potential ratings of users with similar preferences. Thus, it can be used to predict traffic states. Specifically, in the HTMF model, latent factors are employed to capture the global seasonality and trends of urban traffic states. The Hankel structure within HTMF further enhances the local trends observed in time series data. Consequently, the proposed forecasting method can consider both global (long-term) and local (short-term) time series trends of urban traffic states.

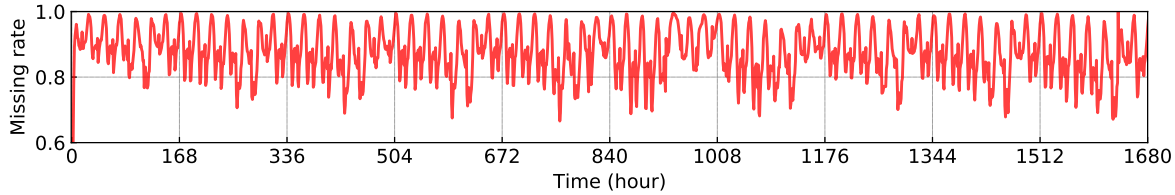
We compare the proposed HTMF model with state-of-the-art baseline models. Our experiments used a hardware setup featuring a T4 GPU with 16GB of RAM. We use the mean absolute percentage error (MAPE) and the root mean square error (RMSE) as the performance metrics, defined as follows,

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

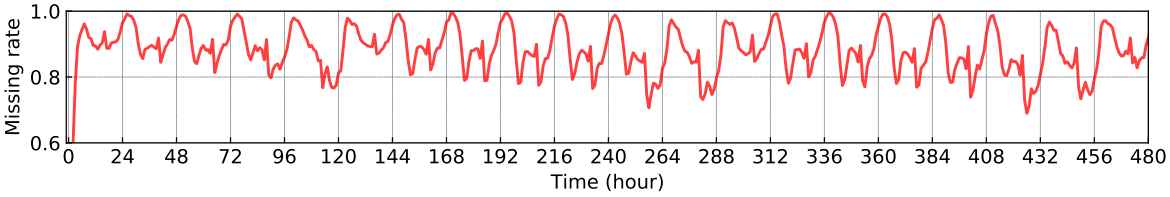
where n is the total number of estimated values, and y_i and $\hat{y}_i$ are the actual value and the estimate, respectively.

4.1. Sparse Urban Traffic State Dataset

The Uber movement speed dataset collects hourly traffic speeds from thousands of road segments in urban areas. However, due to the insufficient sampling and the limited penetration of ridesharing vehicles in total traffic, we only have access to a small fraction of observed speeds, even in an hourly resolution. We use the Seattle Uber movement speed data collected from 63,490 road segments during the first ten weeks of 2019 (i.e., 1680 hours) for the following experiments, and therefore the size of this dataset is 63490×1680. The dataset is sparse and only contains 12.65% speed observations, implying that 87.35% entries in the data matrix are unobserved.



(a) During the ten weeks.



(b) During the first two weeks.

Figure 6 The missing rate curve of the Seattle Uber movement speed dataset.

Figure 6 shows the missing rates of the dataset changing over time during the ten weeks, where each missing rate at time t corresponds to the ratio of missing entries in $\boldsymbol{y}_t$ (the t-th

Table 1 Summary of the ratio of road segments with data with minimum observation rate of the Uber movement speed data sets. # seg. stands for the number of road segments.

Missing rate	# seg.	Ratio	Missing rate	# seg.	Ratio
$(0, 10\%]$	579	0.91%	$(50\%, 60\%]$	1,765	2.78%
$(10\%, 20\%]$	1,273	2.01%	$(60\%, 70\%]$	1,951	3.07%
$(20\%, 30\%]$	1,088	1.71%	$(70\%, 80\%]$	2,687	4.23%
$(30\%, 40\%]$	1,248	1.97%	$(80\%, 90\%]$	5,892	9.28%
$(40\%, 50\%]$	1,629	2.57%	$(90\%, 100\%]$	45,378	71.47%

column of $\boldsymbol{Y}$). It reveals a clear pattern that the dataset has more traffic speed observations during peak hours than during off-peak hours. This is because urban road networks usually serve more vehicles during peak hours. The extreme case is that the missing rates almost reach 100% at midnight, and it seems to be the blackout missing at some consecutive time steps (i.e., corrupted columns). In particular, we summarize the missing rate of the dataset at each road segment in Table 1. The ratio is determined by the division of the number of road segments in a certain missing rate range by the total number of road segments.

From Table 1, we observe that only 2.92% road segments have more than 80% speed observations, i.e., $0.91\% + 2.01\% = 2.92\%$. There are 71.47% road segments that reveal the missing rates greater than 90%. Thus, it poses both practical and methodological challenges for performing time series forecasting on such sparse traffic state data.

In the following experiments, we first evaluate the model with 9-week data as the training set and one-week data as the test set. We set the forecasting time horizons as $\delta = 1, 2, 3, 4$, respectively. Then, to test the model on shorter time series, we set the training set as 8-week data, 7-week data, 6-week data, 5-week data, and 4-week data, respectively. When implementing Algorithm 2, we initialize the factor matrices $\boldsymbol{W}$ and $\boldsymbol{X}$ with small Gaussian random values, i.e., $\frac{1}{100}\mathcal{N}(0, 1)$, where $\mathcal{N}(\cdot)$ denotes the Gaussian distribution.

4.2. Baseline Models

To evaluate the performance of our HTMF model, we compare it with 1) Temporal Regularized Matrix Factorization (TRMF) proposed by Yu et al. (2016), 2) Bayesian TMF (BTMF) proposed by Chen and Sun (2021), 3) Nonstationary TMF (NoTMF) proposed by Chen et al. (2022), and 4) Circulant Tensor Nuclear Norm Minimization (CTNNM) proposed by Liu and Zhang (2022) and Liu (2022). NoTMF integrates the VAR process on seasonal differenced temporal factors into the matrix factorization, which can overcome both non-stationarity and high-dimensionality issues. NoTMF has been empirically demonstrated to

be state-of-the-art over other TMF models such as TRMF and BTMF. Another reason for using the NoTMF as the baseline is that the comparison allows one to reveal the significance of Hankel structure on temporal factors. To reproduce the results, readers can refer to the IJOC GitHub software repository (https://github.com/INFORMSJoC/2022.0197).

4.3. Forecasting Results

We now discuss HTMF's forecasting performance. Since the values of rank R and window length d are critical to HTMF's performance, we first tune these hyperparameters by setting $R = 10, 20, 30, 40, 50, 60$ and $d = 6, 12, 18, 24$. Table 2 reports the results, where the best results are highlighted in bold fonts. It shows that the best forecasting performance is mostly achieved with the rank $R = 60$. In HTMF with a relatively large rank R, the constructed Hankel matrix on latent factors would contribute to more accurate forecasting performance because the complicated time series trends can be characterized by latent factors, including dominant and secondary factors with an appropriate rank R. It should be noted that while it is possible to further increase the value of R, the performance improvement becomes marginal as R increases. Additionally, a higher rank poses a greater risk of overfitting, which can adversely impact the model's performance despite the increased computational cost. Therefore, in practical applications, it is advisable to select an appropriate rank, rather than aiming for the peak value of R. If the forecasting task has a relatively small time horizon (e.g., $\delta = 1$), HTMF with different window lengths shows no significant differences. When the time horizon changes to $\delta = 3$ and 4, HTMF with a longer window length performs significantly better than that with a shorter window length. The window length d of HTMF is significant for capturing time-evolving coefficients of latent factors implicitly, and its setting also depends on forecasting purposes. Based on the results here, we set $R = 60$ in the subsequent experiments. The setting of window length d depends on the forecasting horizon δ, and we can usually set $d > \delta$. Both parameters can be easily validated from a small set of candidate parameters.

In contrast to the autoregressive process, the Hankel structure is an automatic delay embedding transform for time series modeling (Sedighin et al. 2020), which possibly shows superior performance for forecasting short time series (Shi et al. 2020). In our specification, we use training data of different time ranges to test HTMF and the other four baseline models. Table 3 summarizes all the models' forecasting performance with different data ranges, where the best results are highlighted in bold fonts. When using the 9-week

Table 2 Forecasting performance (MAPE/RMSE) of HTMF with different ranks and window lengths.

Horizon	Order	$R=10$	$R=20$	$R=30$	$R=40$	$R=50$	$R=60$
$\delta=1$	$d=6$	10.97/3.47	10.53/3.33	11.04/3.54	10.62/3.40	10.69/3.46	13.44/4.62
	$d=12$	11.25/3.59	10.54/3.36	10.33/3.30	10.25/3.27	10.18/3.23	10.17/3.26
	$d=18$	11.13/3.56	10.50/3.37	10.31/3.32	10.33/3.28	**10.24**/**3.27**	10.26/**3.27**
	$d=24$	11.21/3.58	10.70/3.43	10.42/3.33	10.33/3.31	10.30/3.29	10.26/**3.27**
$\delta=2$	$d=6$	11.37/3.62	15.36/5.29	17.79/6.20	15.10/5.25	13.56/4.71	13.02/4.44
	$d=12$	11.40/3.62	10.73/3.44	10.54/3.38	10.48/3.37	10.54/3.42	10.51/3.42
	$d=18$	11.49/3.65	10.73/3.46	10.51/3.38	10.44/3.32	10.40/3.34	**10.35**/**3.31**
	$d=24$	11.38/3.62	10.78/3.47	10.62/3.40	10.44/3.36	10.53/3.37	10.38/3.32
$\delta=3$	$d=6$	12.37/3.96	14.39/4.96	13.21/4.49	11.75/3.88	13.62/4.59	15.19/5.12
	$d=12$	11.58/3.68	11.87/3.90	10.78/3.53	10.72/3.50	10.98/3.61	11.55/3.82
	$d=18$	11.50/3.64	10.98/3.54	10.84/3.52	10.55/3.40	11.50/3.78	10.67/3.46
	$d=24$	11.31/3.63	10.99/3.54	10.65/3.43	10.52/3.39	**10.49**/**3.38**	10.53/**3.38**
$\delta=4$	$d=6$	12.29/4.09	14.67/4.92	17.34/5.96	19.21/6.51	14.82/4.99	13.93/4.71
	$d=12$	11.50/3.70	11.54/3.84	10.74/3.53	12.29/4.12	11.18/3.74	10.88/3.59
	$d=18$	11.84/3.72	10.83/3.52	10.77/3.55	10.72/3.52	10.52/3.41	10.82/3.53
	$d=24$	11.58/3.66	10.98/3.55	10.84/3.50	10.69/3.45	10.66/3.49	**10.50**/**3.39**

and 8-week data as the training sets, it shows that HTMF achieves competitive results as NoTMF at different time horizons. However, when fewer data are used for training, HTMF significantly outperforms NoTMF in both performance metrics, highlighting the importance of implicit Hankel completion for relatively shorter time series. Notably, in the rolling forecasting process, the forecasting task is converted into a Hankel completion task via the use of matrix factorization with a fixed dictionary matrix $\boldsymbol{U}$. The superiority of HTMF becomes remarkable over NoTMF when the training process only covers 7-week, 6-week, 5-week, and 4-week data. Notably, as the length of training data becomes shorter, HTMF does not show a significant performance loss as NoTMF. These empirically demonstrate that compared to the VAR process in NoTMF, the Hankel structure in HTMF is more effective for handling short time series. Table 3 also clearly shows the superiority of HTMF over TRMF, BTMF, and CTNNM.

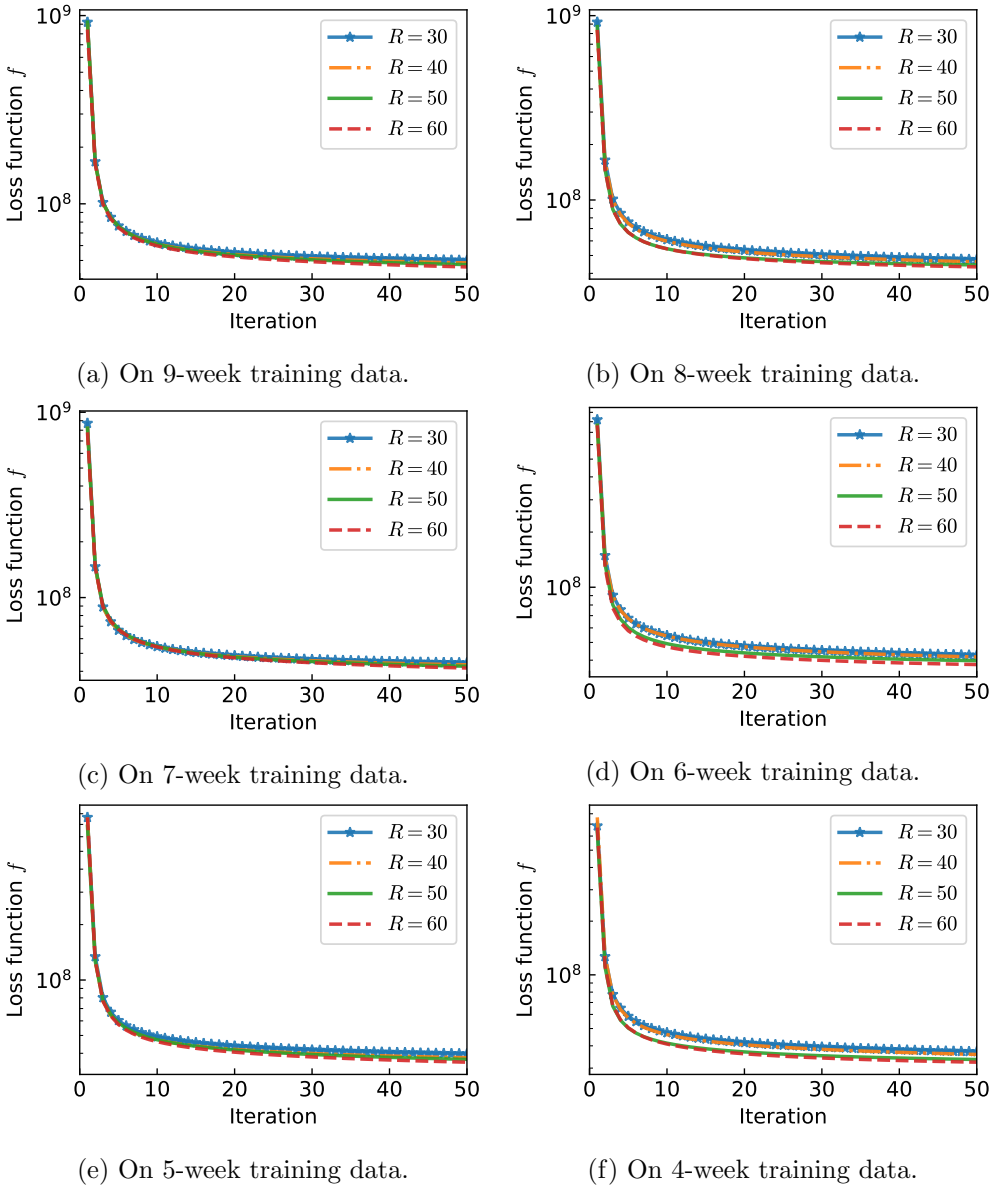
For analyzing the convergence pattern of HTMF, we visualize the loss function values in the iterative process with different ranks (i.e., $R=30,40,50,60$) and different lengths of training data in Figure 7. The figure shows that the loss function of HTMF decreases rapidly in the first few iterations (e.g., smaller than 10 iterations). Then, in the subsequent iterations, the changes in loss function values become marginal. Therefore, the proposed HTMF model shows a very efficient convergence process. Observing the HTMF model with different ranks, the loss function values are relatively smaller when a greater rank is given.

Table 3 Forecasting performance (MAPE/RMSE) of HTMF and baseline models.

	Horizon	9-week	8-week	7-week	6-week	5-week	4-week
HTMF	$\delta=1$	**10.26/3.27**	**10.26/3.26**	**10.31/3.29**	**10.34/3.32**	**10.52/3.37**	**10.50/3.41**
	$\delta=2$	**10.35/3.31**	**10.40/3.32**	**10.49/3.35**	**10.52/3.36**	**10.52/3.39**	**10.63/3.47**
	$\delta=3$	10.53/3.38	**10.46/3.36**	**10.55/3.39**	**10.61/3.40**	**10.64/3.45**	**10.69/3.50**
	$\delta=4$	10.50/3.39	**10.52/3.41**	**10.65/3.43**	**10.66/3.44**	**10.74/3.49**	**10.74/3.52**
NoTMF	$\delta=1$	10.39/3.31	10.54/3.36	10.64/3.41	10.81/3.47	11.16/3.60	11.59/3.76
	$\delta=2$	10.41/3.34	10.52/3.38	10.62/3.42	10.79/3.48	11.12/3.61	11.66/3.82
	$\delta=3$	**10.44/3.35**	10.53/3.41	10.71/3.46	10.83/3.51	11.10/3.61	11.69/3.84
	$\delta=4$	**10.47/3.37**	10.56/3.42	10.68/3.46	10.88/3.54	11.15/3.65	11.45/3.76
TRMF	$\delta=1$	11.10/3.81	11.12/3.86	11.23/3.96	11.29/3.99	11.38/4.02	11.51/4.05
	$\delta=2$	11.03/3.58	10.96/3.59	11.10/3.65	11.20/3.71	11.27/3.75	11.36/3.82
	$\delta=3$	11.54/4.17	11.84/4.31	11.62/4.29	11.65/4.21	11.90/4.38	11.96/4.42
	$\delta=4$	11.07/3.64	11.11/3.67	11.14/3.72	11.31/3.79	11.38/3.83	11.39/3.86
BTMF	$\delta=1$	12.44/4.03	12.31/4.01	12.50/4.06	12.95/4.27	13.16/4.23	12.44/4.00
	$\delta=2$	12.79/4.23	12.59/4.03	12.70/4.10	12.75/4.14	13.44/4.33	12.71/4.09
	$\delta=3$	12.22/3.87	12.12/3.85	12.51/3.96	13.02/4.16	12.87/4.13	12.03/3.88
	$\delta=4$	12.24/3.95	12.23/3.91	12.66/4.04	12.64/4.03	13.09/4.30	12.35/3.95
CTNNM	$\delta=1$	12.74/3.83	12.18/3.71	12.52/3.81	12.76/3.87	13.21/4.03	13.92/4.29
	$\delta=2$	12.97/3.90	12.46/3.78	12.81/3.88	13.16/3.96	13.56/4.08	14.09/4.28
	$\delta=3$	13.36/3.97	12.82/3.85	13.13/3.94	13.66/4.06	13.82/4.12	13.99/4.22
	$\delta=4$	13.58/4.03	13.14/3.92	13.38/3.99	13.73/4.07	13.82/4.14	13.92/4.23

Figure 8 shows the aggregated forecasts achieved by HTMF against the ground truth data with the 9-week training data. The results generated by HTMF are close to the ground truth data, demonstrating the capability of HTMF to accurately forecast the sparse traffic states. The forecasting results show both global (e.g., cyclical patterns) and local (e.g., consistency of traffic flow) time series trends achieved by the Hankel completion in HTMF. Figure 9 plots speed observations and HTMF forecasts of some road segments. Considering the road segments with low missing rates, the forecasts achieved by HTMF capture accurate temporal dynamics, mostly benefiting from the sparse learning process on sparse time series data and the Hankel completion on latent temporal factors within HTMF. For those road segments whose speed observations are very limited, the forecasts achieved by HTMF are consistent with the traffic state fact; namely, speeds are relatively low during peak hours and relatively high during off-peak hours. These findings also reveal the effectiveness of integrating matrix factorization and Hankel representation for identifying temporal dynamics from sparse traffic state data.

Figure 10 shows the statistical histograms of the ground truth data versus the forecasts achieved by HTMF with different missing rate ranges. For each missing rate range, we collect the speed observations of road segments whose missing rate of the test set is in



(a) On 9-week training data.

(b) On 8-week training data.

(c) On 7-week training data.

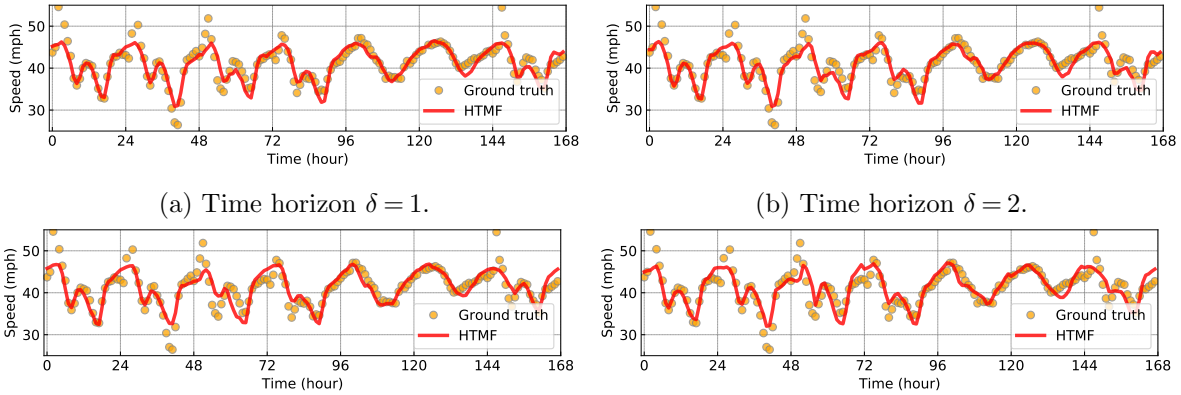
(d) On 6-week training data.

(e) On 5-week training data.

(f) On 4-week training data.

Figure 7 **The variations of HTMF's loss functions with the iterations on different datasets.**

that range. For road segments with lower missing rates, e.g., $(0, 10\%]$ and $(10\%, 20\%]$, the histograms present two peaks. Specifically, one peak is around the speed of 20mph, while another is around the speed of 60mph. This implies that for the road segments with relatively complete speed observations, both lower speeds of the congested traffic (possibly during peak hours) and higher speeds of the free-flow traffic (possibly during off-peak hours) are measured, revealing the bi-mode traffic states. Figure 10 shows that HTMF can accurately forecast both congested and free-flow traffic states. With the increase in missing rates, the phenomena of bi-mode traffic states are weakened. Overall, we can summarize that the forecasts achieved by HTMF are accurate to the ground truth data.



(a) Time horizon $\delta = 1$. (b) Time horizon $\delta = 2$.

(c) Time horizon $\delta = 3$. (d) Time horizon $\delta = 4$.

Figure 8 **Forecasting results of HTMF with different time horizons. Both ground truth data and HTMF forecasts are averaged over 741 road segments whose missing rates on the test set are less than 10%.**

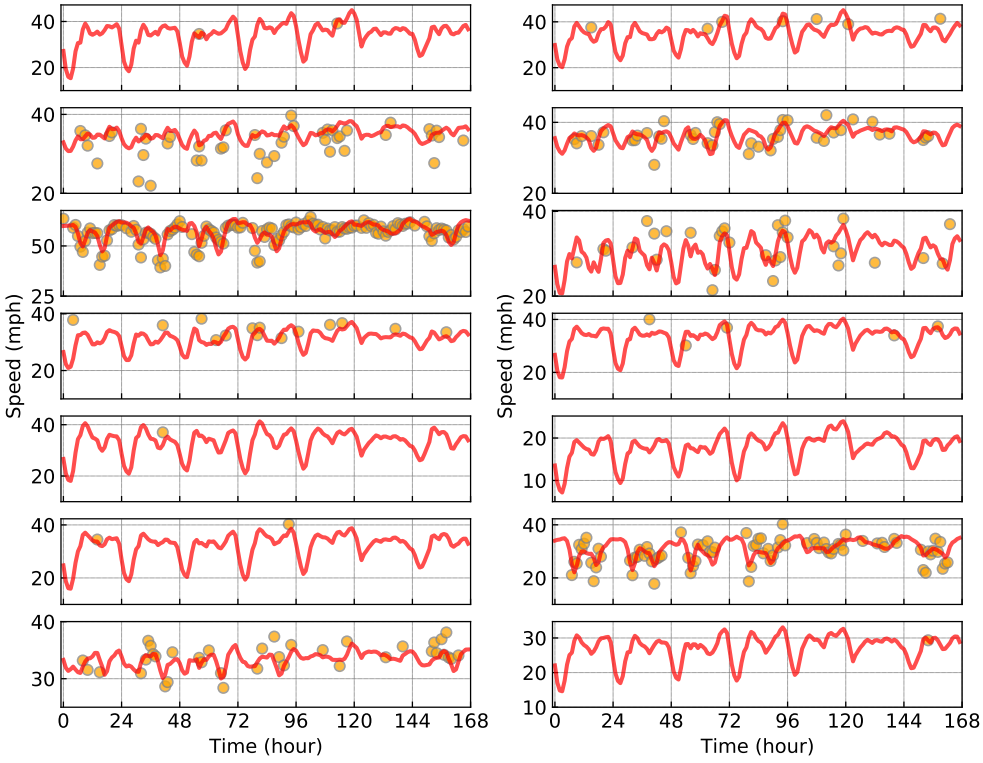
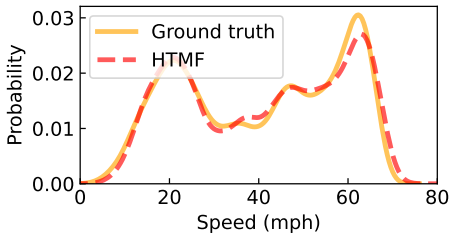


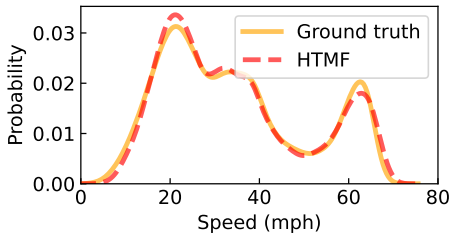
Figure 9 **Forecasting results of HTMF with $\delta = 4$. The 14 example time series correspond to 14 road segments. The orange scatters indicate the observed speeds, while the red curves indicate the HTMF's forecasts.**

5. Conclusion

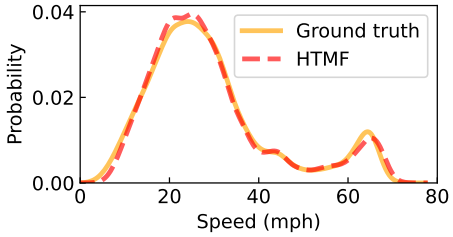
This study revisits the challenging task of urban traffic state forecasting and addresses the sparsity and high-dimensionality issues inherent in city-wide traffic state data. To handle the traffic state data involving multiple characteristics, we present an HTMF model for



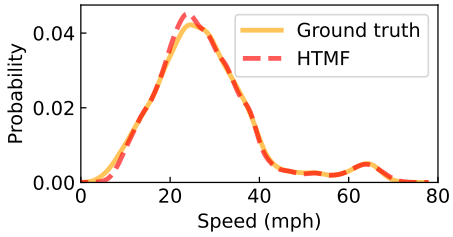
(a) Missing rate $(0, 10\%]$, 741 road segments.



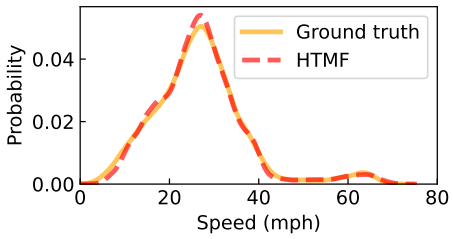
(b) Missing rate $(10\%, 20\%]$, 1,650 road segments.



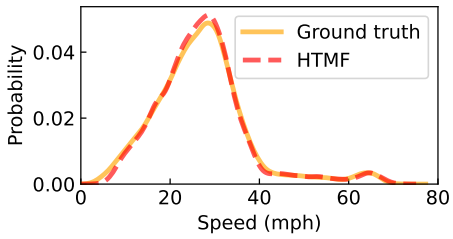
(c) Missing rate $(20\%, 30\%]$, 1,432 road segments.



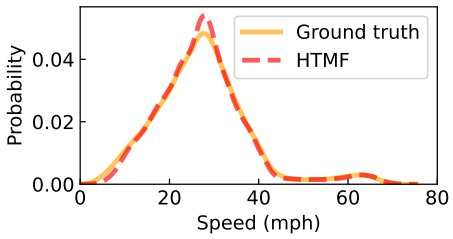
(d) Missing rate $(30\%, 40\%]$, 1,551 road segments.



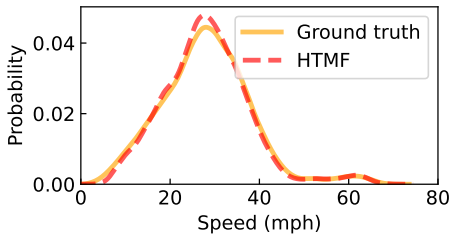
(e) Missing rate $(40\%, 50\%]$, 1,712 road segments.



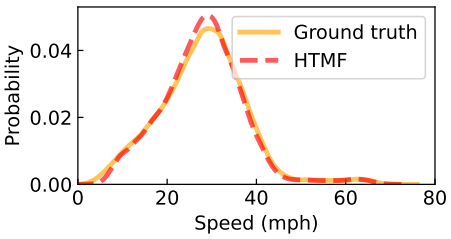
(f) Missing rate $(50\%, 60\%]$, 1,769 road segments.



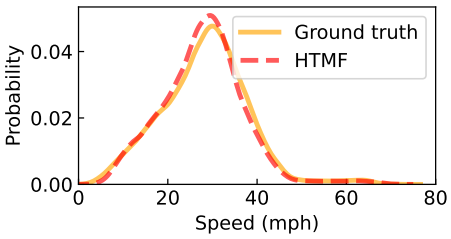
(g) Missing rate $(60\%, 70\%]$, 2,076 road segments.



(h) Missing rate $(70\%, 80\%]$, 3,365 road segments.



(i) Missing rate $(80\%, 90\%]$, 6,602 road segments.



(j) Missing rate $(90\%, 100\%]$, 42,592 road segments.

Figure 10 **The histogram of ground truth data and forecasts achieved by HTMF with $\delta = 4$ in the test set. The missing rate implies the ratio of missing values of road segments in the test set.**

sparse traffic state forecasting. The model consists of spatiotemporal factor matrices in a matrix factorization framework and a Hankel matrix in lower-dimensional spaces. This HTMF model differs from the prior NoTMF model because NoTMF integrates VAR into the seasonal differenced temporal factors. In contrast, HTMF utilizes a Hankel matrix on the lower-dimensional temporal factors to achieve temporal modeling. In this situation, HTMF does not require the stationarity of traffic states.

In order to effectively leverage real-time incremental data and achieve efficient forecasting, the proposed forecasting mechanism encompasses two key processes: online imputation and online forecasting. In particular, online imputation reconstructs missing values in the incremental data, while online forecasting utilizes Hankel matrix factorization to estimate future traffic states. Notably, the approach is efficient and effective due to using the Hankel matrix in the lower-dimensional latent spaces. The Hankel matrix can implicitly preserve the temporal correlations for traffic states and provide insight into estimating future traffic states automatically. Extensive numerical experiments were conducted using the real-world Uber movement speed dataset from Seattle, USA, to validate the proposed methodology. The results demonstrate the superior forecasting performance of the HTMF model compared to multiple baseline models. These findings provide compelling evidence for the advantages of HTMF in effectively addressing challenges associated with sparsity, high-dimensionality, and short time series in urban traffic state forecasting. Our proposed HTMF model also holds potential for other prediction tasks as long as the real-world time series forecasting exhibits two key characteristics: sparse time series with global/local trends and low-dimensional patterns. These characteristics indicate that the high-dimensional time series can be effectively compressed or represented using limited low-dimensional factors and simultaneously take the expression of matrix factorization on the sparse data.

In this work, we empirically analyzed the convergence pattern of HTMF. Although the convergence behavior of an algorithm is often affected by several factors, such as the assumptions made in the model and the methods for initialization (Chi et al. 2019), it is still crucial to analyze the convergence behavior of the algorithm with theoretical guarantees. Thus, a potential direction for advancing this study is to investigate whether theoretical performance guarantees can be derived under certain assumptions and specific initialization schemes.

Acknowledgments

This research was supported by the Institute for Data Valorisation (IVADO), the Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), the National Natural Science Foundation of China [Grants 12371456, 72101049, 72232001], the Sichuan Science and Technology Program [Grant 2024NSFJQ0038], and the Fundamental Research Funds for the Central Universities [Grant DUT23RC(3)045]. The authors thank the associate editor and anonymous referees for their insightful comments.

References

Anava, Oren, Elad Hazan, Assaf Zeevi. 2015. Online time series prediction with missing data. *International Conference on Machine Learning*. PMLR, 2191–2199.

Cai, Jian-Feng, Tianming Wang, Ke Wei. 2019. Fast and provable algorithms for spectrally sparse signal reconstruction via low-rank hankel matrix completion. *Applied and Computational Harmonic Analysis* **46**(1) 94–121.

Che, Zhengping, Sanjay Purushotham, Kyunghyun Cho, David Sontag, Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports* **8**(1) 1–12.

Chen, Jinchi, Weiguo Gao, Ke Wei. 2021. Exact matrix completion based on low rank hankel structure in the fourier domain. *Applied and Computational Harmonic Analysis* **55** 149–184.

Chen, Xinyu, Zhaocheng He, Lijun Sun. 2019. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies* **98** 73–84.

Chen, Xinyu, Lijun Sun. 2021. Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**(9) 4659–4673.

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

Chen, Xinyu, Chengyuan Zhang, Xi-Le Zhao, Nicolas Saunier, Lijun Sun. 2022. Nonstationary temporal matrix factorization for multivariate time series forecasting. *arXiv preprint arXiv:2203.10651* .

Chen, Yudong, Yuejie Chi. 2018. Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization. *IEEE Signal Processing Magazine* **35**(4) 14–31.

Chi, Yuejie, Yue M. Lu, Yuxin Chen. 2019. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing* **67**(20) 5239–5269.

Furuhata, Masabumi, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, Sven Koenig. 2013. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological* **57** 28–46.

Golub, Gene H., Charles F. Van Loan. 2013. *Matrix Computations*. 4th ed. The Johns Hopkins University Press.

Gultekin, San, John Paisley. 2018. Online forecasting matrix factorization. *IEEE Transactions on Signal Processing* **67**(5) 1223–1236.

Hamilton, James Douglas. 1994. *Time series analysis*. Princeton university press.

Hu, Xinyu, Olcay Cirit, Tanmay Binaykiya, Ramit Hora. 2022. Deepeta: How Uber predicts arrival times using deep learning. Accessed October 26, 2023, https://www.uber.com/en-HK/blog/deepeta-how-uber-predicts-arrival-times/.

Koren, Yehuda, Robert Bell, Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* **42**(8) 30–37.

Li, Li, Yuebiao Li, Zhiheng Li. 2013. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. *Transportation Research Part C: Emerging Technologies* **34** 108–120.

Lim, Bryan, Stefan Zohren. 2021. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A* **379**(2194) 20200209.

Liu, Guangcan. 2022. Time series forecasting via learning convolutionally low-rank models. *IEEE Transactions on Information Theory* **68**(5) 3362–3380.

Liu, Guangcan, Wayne Zhang. 2022. Recovery of future data via convolution nuclear norm minimization. *IEEE Transactions on Information Theory* **69**(1) 650–665.

Liu, Siyuan, Shaojie Tang, Jiangchuan Zheng, Lionel M Ni. 2022. Unsupervised learning for human mobility behaviors. *INFORMS Journal on Computing* **34**(3) 1565–1586.

Mnih, Andriy, Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems* **20**.

Qin, Huiling, Xianyuan Zhan, Yuanxun Li, Xiaodu Yang, Yu Zheng. 2021. Network-wide traffic states imputation using self-interested coalitional learning. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1370–1378.

Rao, Nikhil, Hsiang-Fu Yu, Pradeep K Ravikumar, Inderjit S Dhillon. 2015. Collaborative filtering with graph information: Consistency and scalable methods. *Advances in neural information processing systems* **28**.

Rodrigues, Filipe, Kristian Henrickson, Francisco C Pereira. 2018. Multi-output gaussian processes for crowdsourced traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* **20**(2) 594–603.

Sedighin, Farnaz, Andrzej Cichocki, Tatsuya Yokota, Qiquan Shi. 2020. Matrix and tensor completion in multiway delay embedded space using tensor train, with application to signal reconstruction. *IEEE Signal Processing Letters* **27** 810–814.

Sen, Rajat, Hsiang-Fu Yu, Inderjit S Dhillon. 2019. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in Neural Information Processing Systems* **32**.

Shi, Qiquan, Jiaming Yin, Jiajun Cai, Andrzej Cichocki, Tatsuya Yokota, Lei Chen, Mingxuan Yuan, Jia Zeng. 2020. Block hankel tensor arima for multiple short time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34. 5758–5766.

Tang, Xianfeng, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, Suhang Wang. 2020. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34. 5956–5963.

Treiber, Martin, Arne Kesting. 2013. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg* 983–1000.

Xiong, Liang, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 211–222.

Yokota, Tatsuya, Burak Erem, Seyhmus Guler, Simon K Warfield, Hidekata Hontani. 2018. Missing slice recovery for tensors using a low-rank model in embedded space. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8251–8259.

Yu, Hsiang-Fu, Nikhil Rao, Inderjit S Dhillon. 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in Neural Information Processing Systems* **29** 847–855.

Zhang, Shuai, Yingshuai Hao, Meng Wang, Joe H Chow. 2018. Multichannel hankel matrix completion through nonconvex optimization. *IEEE Journal of Selected Topics in Signal Processing* **12**(4) 617–632.

Zhang, Shuai, Meng Wang. 2019. Correction of corrupted columns through fast robust hankel matrix completion. *IEEE Transactions on Signal Processing* **67**(10) 2580–2594.

Zhang, Yanqing, Xuan Bi, Niansheng Tang, Annie Qu. 2021. Dynamic tensor recommender systems. *The Journal of Machine Learning Research* **22**(1) 3032–3066.

Zheng, Yu. 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* **6**(3) 1–41.