



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE



Matrix and Tensor Models for Spatiotemporal Traffic Data Imputation and Forecasting

Xinyu Chen

Ph.D., University of Montreal, Canada

December 28, 2023



Southern University
of Science and
Technology



Outline

1. Background

- Spatiotemporal traffic data
- Urban movement data
- Problem formulation

2. Literature Review

- Tensor factorization
- Matrix/Tensor completion

3. Nonstationary Temporal Matrix Factorization (NoTMF)

- Model description
- Sparse urban traffic state forecasting

4. Laplacian Convolutional Representation (LCR)

- Model description
- Univariate traffic time series imputation
- Large-scale traffic data imputation

5. Hankel Tensor Factorization (HTF)

- Model description
- Extreme missing traffic data imputation

6. Conclusion

Spatiotemporal Traffic Data

Many spatiotemporal traffic time series data are in the form of **matrix**.

- Portland highway traffic data¹



- $X \in \mathbb{R}^{N \times T}$ with N spatial locations $\times T$ time steps
- Traffic volume/speed shows strong spatial/temporal dependencies

¹<https://portal.its.pdx.edu/home>

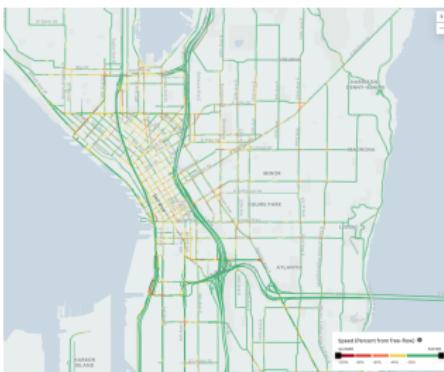
Urban Movement Data

High-dimensional & sparse

- Uber (hourly) movement speed data



NYC movement



Seattle movement

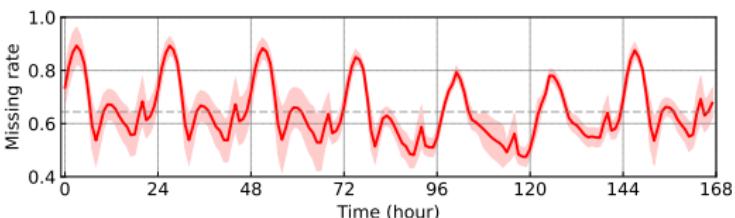
- {road segment, time slot (hour), average speed}
- Computing hourly speed: Road segments have 5+ unique trips.

Issue: Insufficient sampling of ridesharing vehicles on the road network!

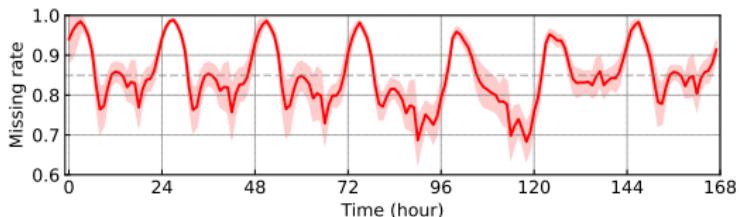
Urban Movement Data

High-dimensional & sparse

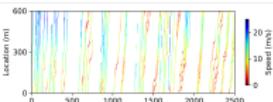
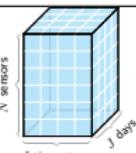
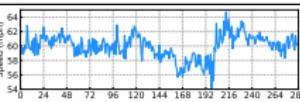
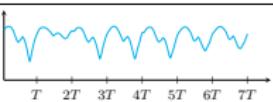
- **NYC** movement speed data (2019)
 - 98,210 road segments & 8,760 time steps (hours)
 - Overall missing rate: 64.43%



- **Seattle** movement speed data (2019)
 - 63,490 road segments & 8,760 time steps (hours)
 - Overall missing rate: 84.95%



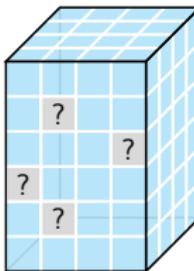
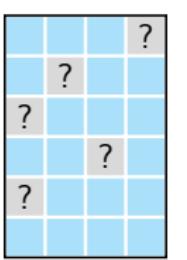
Spatiotemporal Traffic Data

Sparse	
High-dimensional	
Multidimensional	
Noises & outliers	
Nonstationary	

Traffic data show complicated spatiotemporal patterns and correlations.

Problem Formulation

Objective A: Impute missing values in the data matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$ (or tensor $\mathcal{Y} \in \mathbb{R}^{M \times N \times T}$).



- Matrix completion (Observed index set Ω)

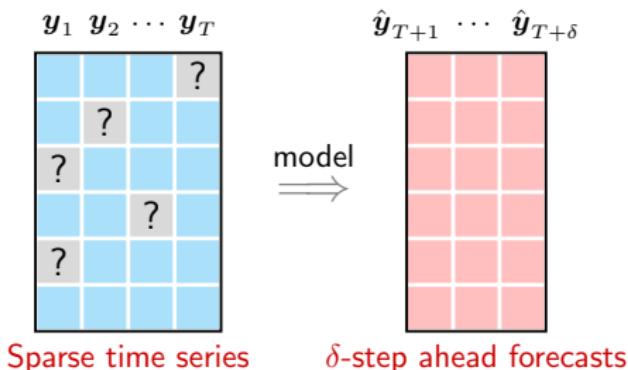
$$\underbrace{\mathcal{P}_\Omega(\mathbf{Y})}_{\text{Partially observed}} \xrightarrow{\text{Estimate}} \underbrace{\mathcal{P}_\Omega^\perp(\mathbf{Y})}_{\text{Unobserved}}$$

Modeling process:

- How to make use of spatiotemporal traffic patterns?
- How to make use of traffic time series dynamics?

Problem Formulation

Objective B: Given a partially observed data $\mathbf{Y} \in \mathbb{R}^{N \times T}$ consisting of time series $\mathbf{y}_1, \dots, \mathbf{y}_T \in \mathbb{R}^N$, forecast data points $\hat{\mathbf{y}}_{T+\delta}, \delta \in \mathbb{N}^+$.



Modeling process:

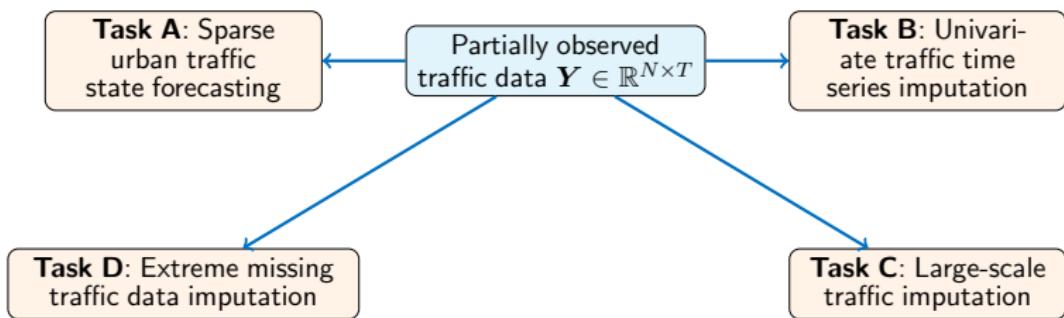
- How to characterize time series dynamics in high-dimensional and sparse traffic data?

Real-world applications:

- Forecasting urban traffic states with sparse data.

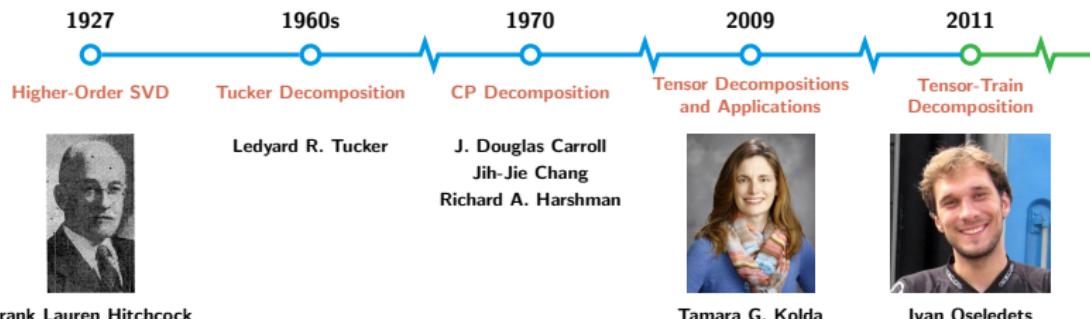
Tasks

Focus: spatiotemporal traffic data imputation and forecasting.



Tensor Factorization

- Revisit tensor factorization

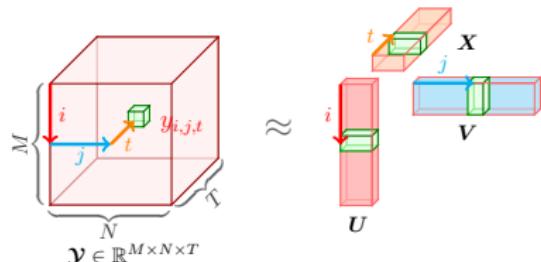


Frank Lauren Hitchcock

Tamara G. Kolda

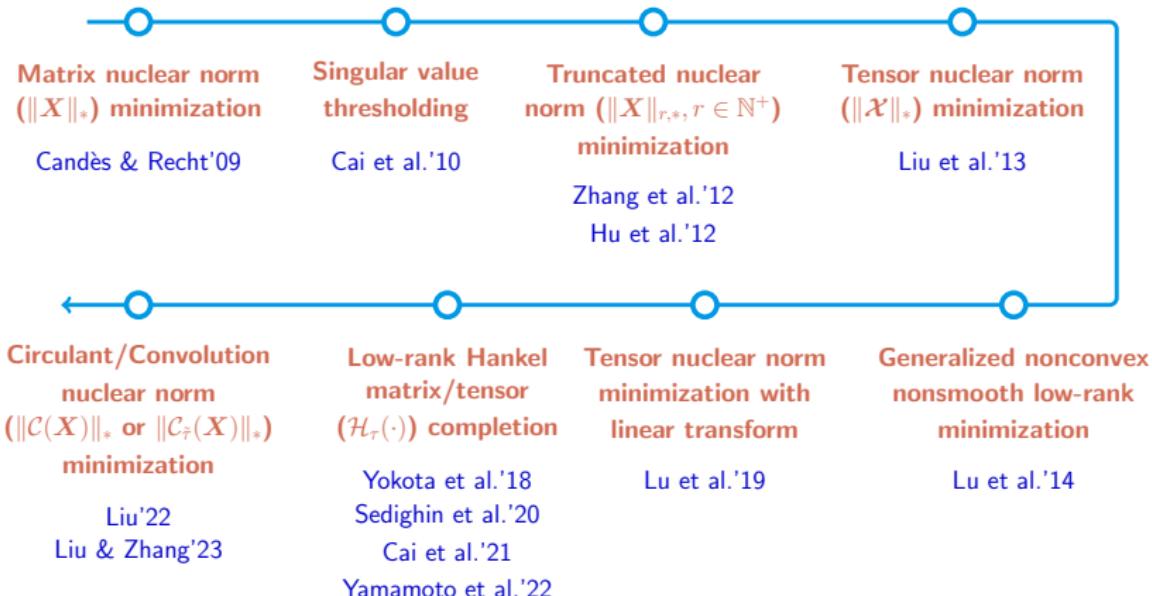
Ivan Oseledets

- CP tensor factorization:** Factorize \mathcal{Y} into the combination of three rank- R factor matrices (i.e., low-dimensional latent factors).



$$\left\{ \begin{array}{l} y_{i,j,t} \approx \sum_{r=1}^R u_{i,r} v_{j,r} x_{t,r} \\ \mathcal{Y} \approx \sum_{r=1}^R \mathbf{u}_r \otimes \mathbf{v}_r \otimes \mathbf{x}_r \end{array} \right.$$

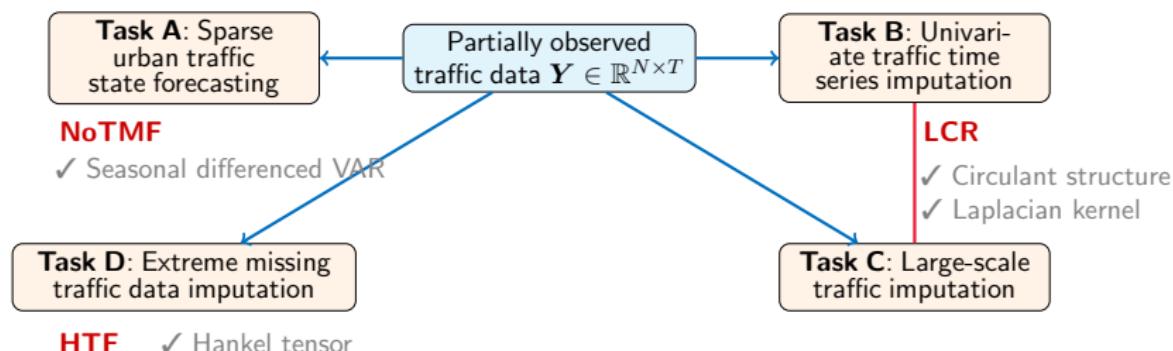
Matrix/Tensor Completion



Overview

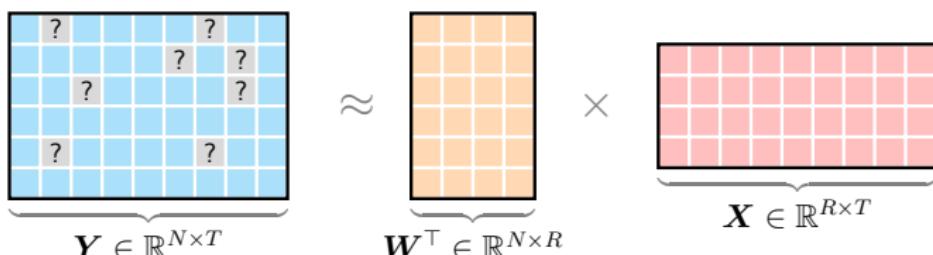
Machine learning framework

- **Matrix and tensor methods:** Learning from sparse data.
- **Temporal modeling:** Building and reinforcing temporal dependencies for time series.



Matrix Factorization

A simple approach to reconstruct missing values.



MF (Koren et al.'09)

Estimating low-dimensional \mathbf{W}, \mathbf{X} :

$$\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2$$

on data \mathbf{Y} w/ observed index set Ω .

- ✓ Learn from sparse data
- ✓ Spatial factor matrix \mathbf{W}
- ✓ Temporal factor matrix \mathbf{X}

How to build temporal correlations on MF?

Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.

$$\begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix} \underbrace{\quad}_{\mathbf{Y} \in \mathbb{R}^{N \times T}} \approx \begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix} \underbrace{\quad}_{\mathbf{W}^\top \in \mathbb{R}^{N \times R}} \times \begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix} \underbrace{\quad}_{\mathbf{X} \in \mathbb{R}^{R \times T}}$$

$$\begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix} \underbrace{\quad}_{\mathbf{Y} \in \mathbb{R}^{N \times T}} \approx \begin{matrix} \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \\ \text{?} & \text{?} \end{matrix} \underbrace{\quad}_{\mathbf{W}^\top \in \mathbb{R}^{N \times R}} \times \begin{matrix} \mathbf{x}_{t-2} & \mathbf{x}_{t-1} & \mathbf{x}_t & \mathbf{x}_{t+1} & \mathbf{x}_{t+2} \\ \dots & & & & \dots \end{matrix} \left. \right\} R$$

time step

\Downarrow **\mathbf{X} is time series?**

Why? Temporal factor matrix $\mathbf{X} \in \mathbb{R}^{R \times T}$ is the low-dimensional representation of time series dynamics of $\mathbf{Y} \in \mathbb{R}^{N \times T}$.

Temporal Matrix Factorization

Vector autoregression (VAR) on the temporal factor matrix.

MF (Koren et al.'09)

Estimating low-dimensional \mathbf{W}, \mathbf{X} :

$$\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2$$

on data \mathbf{Y} w/ observed index set Ω .

dth-order VAR

$$\mathbf{x}_t = \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} + \underbrace{\boldsymbol{\epsilon}_t}_{\mathcal{N}(\mathbf{0}, \mathbf{I})}$$

w/ coefficients $\{\mathbf{A}_k\}$.

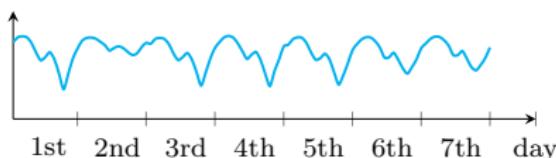


Yu et al.'16
Chen & Sun'21

$$\min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k=1}^d} \frac{1}{2} \underbrace{\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2}_{\text{MF on data } \mathbf{Y}} + \frac{\gamma}{2} \underbrace{\sum_{t=d+1}^T \left\| \mathbf{x}_t - \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} \right\|_2^2}_{\text{VAR on temporal factors } \mathbf{X}}$$

Nonstationary Temporal Matrix Factorization

Traffic data are nonstationary due to daily patterns of traffic flow.



- Season- m differencing ($m \in \mathbb{N}^+$, e.g., daily/weekly):

$$\mathbf{x}_t \approx \sum_{k=1}^d \mathbf{A}_k \mathbf{x}_{t-k} \quad \Rightarrow \quad \mathbf{x}_t - \mathbf{x}_{t-m} \approx \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k})$$

- (Ours) Optimization problem:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k=1}^d} & \frac{1}{2} \underbrace{\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2}_{\text{MF on data } \mathbf{Y}} + \frac{\rho}{2} \underbrace{(\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2)}_{\text{Regularization}} \\ & + \frac{\gamma}{2} \underbrace{\sum_{t=d+m+1}^T \left\| (\mathbf{x}_t - \mathbf{x}_{t-m}) - \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k}) \right\|_2^2}_{\text{VAR on seasonal differenced temporal factors}} \end{aligned}$$

Nonstationary Temporal Matrix Factorization

Rewrite NoTMF

- Optimization problem:²

$$\min_{\mathbf{W}, \mathbf{X}, \mathbf{A}} \frac{1}{2} \underbrace{\|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2}_{\text{MF on data } \mathbf{Y}} + \frac{\rho}{2} \underbrace{(\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2)}_{\text{Regularization}} + \frac{\gamma}{2} \underbrace{\|\mathbf{X}\Psi_0^\top - \mathbf{A}(\mathbf{I}_d \otimes \mathbf{X})\Psi^\top\|_F^2}_{\text{VAR on } \mathbf{X}}$$

where $\Psi_0 \in \mathbb{R}^{(T-d-m) \times T}$, $\Psi \in \mathbb{R}^{(T-d-m) \times (dT)}$ are temporal operators.

- Alternating minimization (let f be the obj.):

$$\left\{ \begin{array}{ll} \text{Spatial factors} & \mathbf{W} := \{\mathbf{W} \mid \frac{\partial f}{\partial \mathbf{W}} = \mathbf{0}\} \quad (\text{least squares}) \\ \text{Temporal factors} & \mathbf{X} := \{\mathbf{X} \mid \frac{\partial f}{\partial \mathbf{X}} = \mathbf{0}\} \quad (\text{conjugate gradient}) \\ \text{VAR coefficients} & \mathbf{A} := \{\mathbf{A} \mid \frac{\partial f}{\partial \mathbf{A}} = \mathbf{0}\} \quad (\text{least squares}) \end{array} \right.$$

² $\mathbf{A} \triangleq [\mathbf{A}_1 \quad \cdots \quad \mathbf{A}_d] \in \mathbb{R}^{R \times (dR)}$ (coefficient matrix).

Nonstationary Temporal Matrix Factorization

NoTMF forecasting?

Implementation

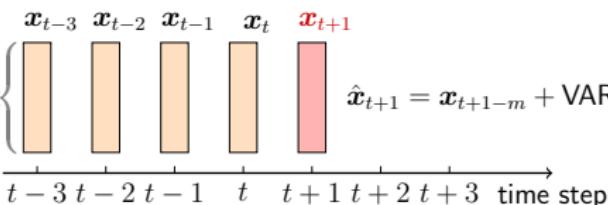
- Estimate $\mathbf{W}, \mathbf{X}, \mathbf{A}$
- Forecast $\hat{\mathbf{x}}_{t+1}$ with VAR
- Return $\hat{\mathbf{y}}_{t+1} = \mathbf{W}^\top \hat{\mathbf{x}}_{t+1}$

- ✓ Sparse input \mathbf{Y}_t
- ✓ Forecast in latent spaces

$$\underbrace{\mathbf{Y}_t \in \mathbb{R}^{N \times t}}_{\text{Matrix } \mathbf{Y}_t \text{ with } N \text{ rows and } t \text{ columns}} \quad \begin{matrix} ? & & & ? & ? \\ & ? & & & ? \\ ? & & & ? & ? \end{matrix}$$

$$R \left\{ \begin{matrix} \mathbf{x}_{t-3} & \mathbf{x}_{t-2} & \mathbf{x}_{t-1} & \mathbf{x}_t & \mathbf{x}_{t+1} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \end{matrix} \right. \quad \hat{\mathbf{x}}_{t+1} = \mathbf{x}_{t+1-m} + \text{VAR}(d, m)$$

time step



Nonstationary Temporal Matrix Factorization

NoTMF forecasting on streaming data?

- Online forecasting (Gultekin & Paisley'18):
 - Fix the spatial factor matrix \mathbf{W}
 - Use input data \mathbf{Y}_{t+1} to update the temporal factor matrix \mathbf{X} and the coefficient matrix \mathbf{A}

Implementation

- Estimate \mathbf{X}, \mathbf{A}
- Forecast $\hat{\mathbf{x}}_{t+2}$ with VAR
- Return $\hat{\mathbf{y}}_{t+2} = \mathbf{W}^\top \hat{\mathbf{x}}_{t+2}$

- ✓ Sparse input \mathbf{Y}_{t+1}
- ✓ Forecast in latent spaces

$$\mathbf{Y}_{t+1} \in \mathbb{R}^{N \times (t+1)}$$

A 4x(t+1) grid of cells. The last column is red and the rest are blue. Cells in the last column are marked with question marks.

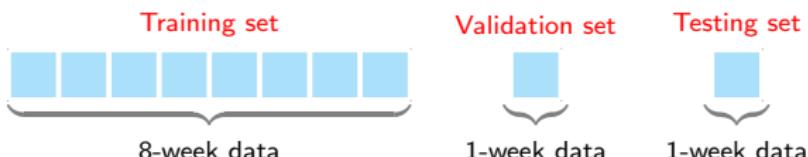
$$R \left\{ \begin{array}{c} \mathbf{x}_{t-3} \quad \mathbf{x}_{t-2} \quad \mathbf{x}_{t-1} \quad \mathbf{x}_t \quad \mathbf{x}_{t+1} \quad \mathbf{x}_{t+2} \\ | \qquad | \qquad | \qquad | \qquad | \qquad | \\ t-3 \quad t-2 \quad t-1 \quad t \quad t+1 \quad t+2 \end{array} \right. \quad \hat{\mathbf{y}}_{t+2} = \mathbf{x}_{t+2-m} + \text{VAR}(d, m)$$

A diagram showing a sequence of temporal factors $\mathbf{x}_{t-3}, \mathbf{x}_{t-2}, \mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{x}_{t+1}, \mathbf{x}_{t+2}$ over time steps $t-3, t-2, t-1, t, t+1, t+2$. The time step $t+2$ is highlighted in red. The formula $\hat{\mathbf{y}}_{t+2} = \mathbf{x}_{t+2-m} + \text{VAR}(d, m)$ is shown below.

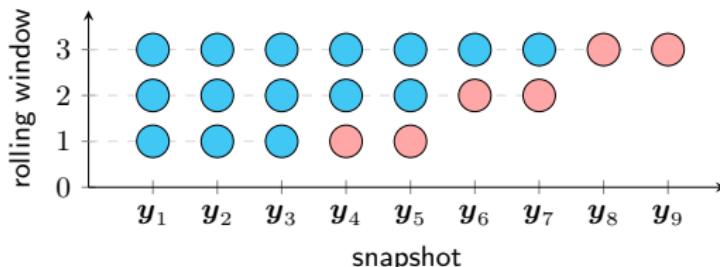
Sparse Urban Traffic State Forecasting

NoTMF forecasting

- NYC Uber movement speed dataset:
 - 10-week data of size 98210×1680 ; **66.56%** missing values
- Rolling forecasting setup (Time horizon $\delta = 1, 2, 3, 6$):



- Weight parameter $\gamma \in \{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\}$
- Weight parameter $\rho = \{10^{-1}\gamma, 5 \times 10^{-1}\gamma, \gamma, 5\gamma, 10\gamma\}$
- Rolling forecasting illustration ($\delta = 2$):



Sparse Urban Traffic State Forecasting

NoTMF vs. baseline (in MAPE/RMSE)

- On the NYC Uber movement speed dataset

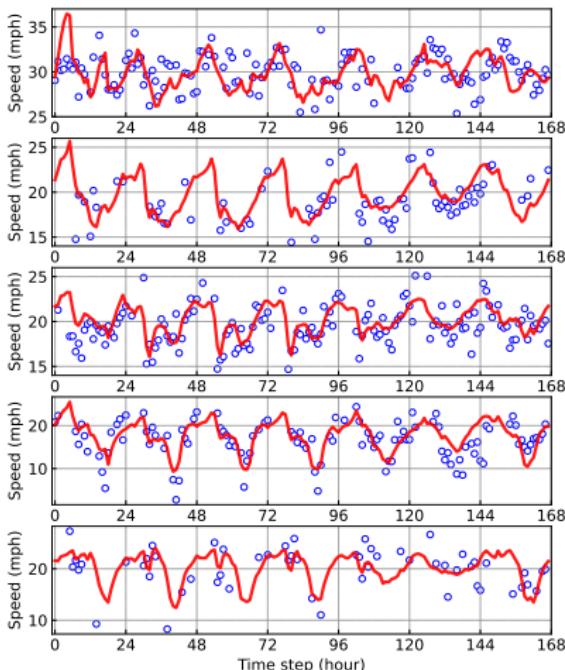
δ	d	NoTMF ($m = 24$)	NoTMF ($m = 168$)	NoTMF-1st ($m = 168$)	TRMF	BTMF	BTRMF
1	1	13.63/2.88	13.53/2.86	13.45/2.85	14.50/3.12	14.94/3.13	15.93/3.33
	2	13.47/2.84	13.41/2.84	13.42/2.84	14.14/3.05	15.70/3.41	15.90/3.35
	3	13.46/2.84	13.39/2.83	13.43/2.84	13.87/2.96	15.80/3.34	16.08/3.43
	6	13.41/2.83	13.39/2.83	13.41/2.83	14.00/2.98	15.45/3.27	16.26/3.48
2	1	13.91/2.96	13.76/2.94	13.70/2.92	15.85/3.43	15.33/3.21	16.85/3.56
	2	13.77/2.92	13.63/2.89	13.72/2.92	15.04/3.31	15.87/3.32	17.27/3.71
	3	13.72/2.91	13.61/2.89	13.73/2.92	15.25/3.36	15.69/3.33	17.24/3.74
	6	13.59/2.87	13.57/2.88	13.68/2.91	14.92/3.24	15.91/3.39	18.18/3.97
3	1	14.30/3.05	14.06/3.02	14.02/3.00	17.52/3.83	15.86/3.32	18.61/3.91
	2	14.01/2.98	13.84/2.94	13.96/2.98	17.32/4.00	16.30/3.40	18.90/4.10
	3	13.95/2.97	13.79/2.93	13.98/2.98	16.91/3.71	16.56/3.49	18.68/4.05
	6	13.78/2.92	13.73/2.92	13.91/2.96	16.72/3.65	15.49/3.27	20.45/4.66
6	1	14.61/3.11	14.67/3.20	14.98/3.32	21.20/4.70	15.99/3.32	22.40/4.69
	2	14.30/3.03	14.33/3.09	14.90/3.28	20.87/5.01	16.04/3.33	23.56/5.63
	3	14.26/3.03	14.28/3.09	14.86/3.26	20.08/4.65	15.67/3.28	24.27/5.72
	6	14.06/2.97	14.16/3.06	14.80/3.23	20.40/4.35	16.38/3.50	26.34/6.60

NoTMF performs better than other TMF models (e.g., TRMF, BTMF & BTRMF).

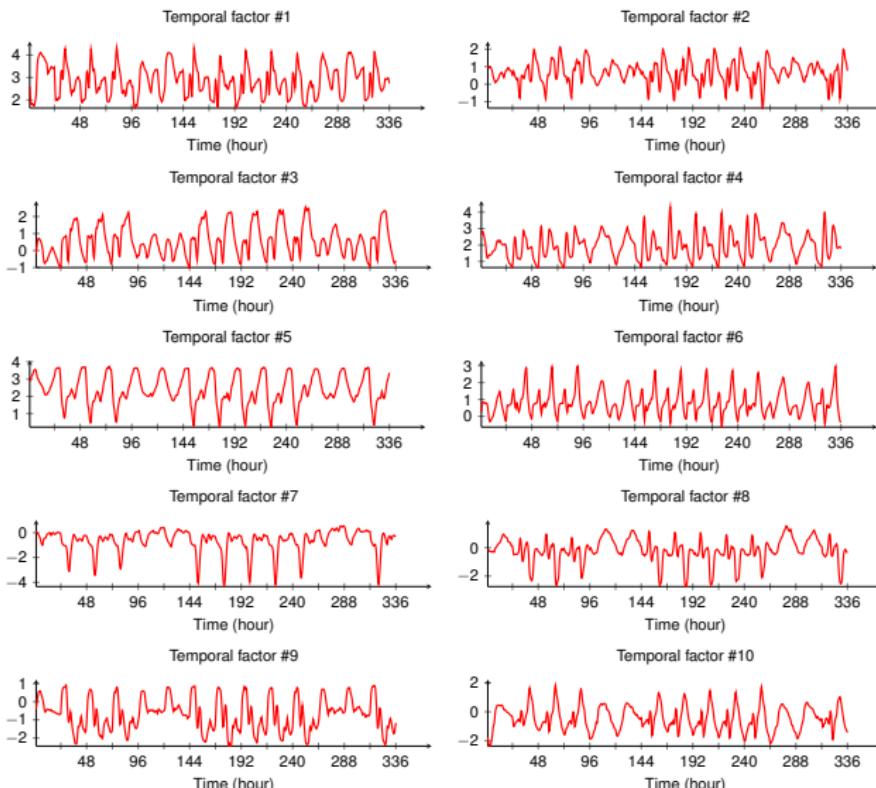
Sparse Urban Traffic State Forecasting

NoTMF forecasting ($\delta = 6$)

- On the NYC Uber movement speed dataset



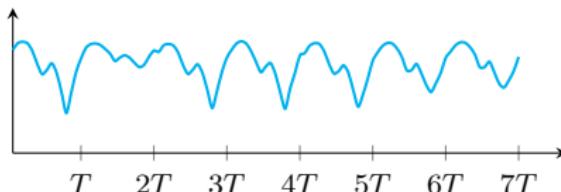
Sparse Urban Traffic State Forecasting



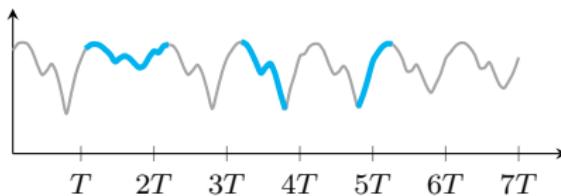
Laplacian Convolutional Representation

Motivation: Time series imputation

- Global trends (e.g., long-term quasi-seasonality & daily/weekly rhythm):



- Local trends (e.g., short-term time series trends):

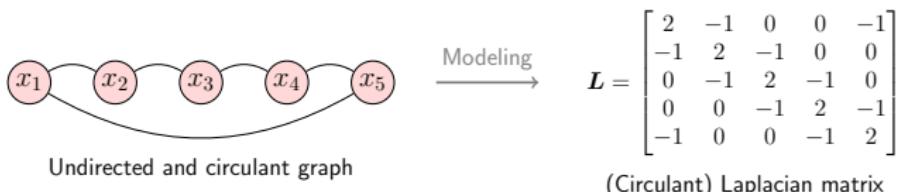


How to characterize both global and local trends in sparse time series?

Laplacian Convolutional Representation

Local trend modeling

- Intuition of (circulant) Laplacian matrix



- Define Laplacian kernel:

$$\ell \triangleq (2, -1, 0, 0, -1)^\top$$

↓

$$\ell \triangleq (\underbrace{2\tau}_{\text{degree}}, \underbrace{-1, \dots, -1}_{\tau}, 0, \dots, 0, \underbrace{-1, \dots, -1}_{\tau})^\top \in \mathbb{R}^T$$

for any time series $\mathbf{x} = (x_1, \dots, x_T)^\top \in \mathbb{R}^T$.

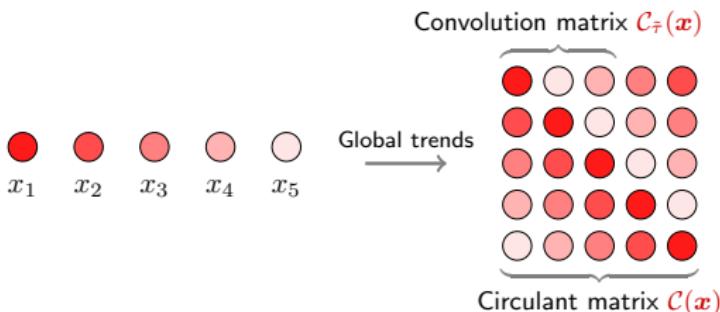
- (Laplacian) Temporal regularization:

$$\mathcal{R}_\tau(\mathbf{x}) = \frac{1}{2} \|\mathbf{L}\mathbf{x}\|_2^2 = \frac{1}{2} \|\ell * \mathbf{x}\|_2^2$$

Reformulate temporal regularization with circular convolution.

Laplacian Convolutional Representation

Global trend modeling: Circulant matrix $\mathcal{C}(\mathbf{x})$ vs. convolution matrix $\mathcal{C}_{\tilde{\tau}}(\mathbf{x})$



- Circulant/Convolution nuclear norm minimization
 - A balance between global and local trends modeling?

CircNNM (Liu'22, Liu & Zhang'23)

Estimating \mathbf{x} :

$$\min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_*$$

$$\text{s.t. } \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon$$

on data \mathbf{y} w/ observed index set Ω .

ConvNNM (Liu'22, Liu & Zhang'23)

Estimating \mathbf{x} :

$$\min_{\mathbf{x}} \|\mathcal{C}_{\tilde{\tau}}(\mathbf{x})\|_*$$

$$\text{s.t. } \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon$$

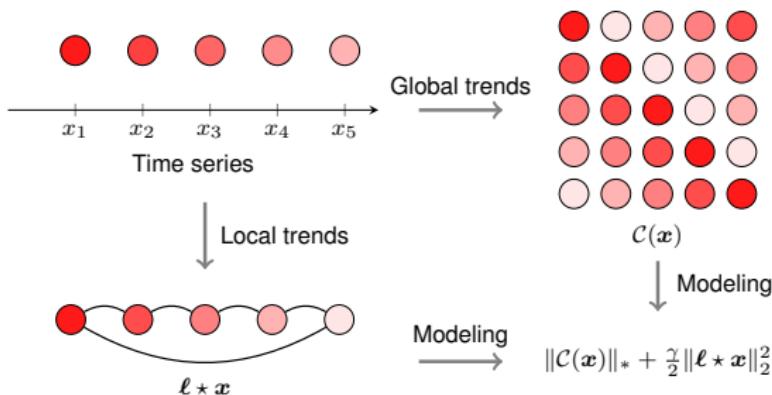
on data \mathbf{y} w/ observed index set Ω .

Laplacian Convolutional Representation

Laplacian Convolutional Representation (LCR)

For any partially observed time series $\mathbf{y} \in \mathbb{R}^T$ with observed index set Ω , LCR utilizes **circulant matrix** and **Laplacian kernel** to characterize global and local trends in time series, respectively, i.e.,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \underbrace{\|\mathcal{C}(\mathbf{x})\|_*}_{\text{global}} + \frac{\gamma}{2} \underbrace{\|\ell * \mathbf{x}\|_2^2}_{\text{local}} \\ \text{s.t. } & \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon \end{aligned}$$



Laplacian Convolutional Representation

- Augmented Lagrangian function:³

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) = \|\mathcal{C}(\mathbf{x})\|_* + \frac{\gamma}{2} \|\boldsymbol{\ell} * \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 + \langle \mathbf{w}, \mathbf{x} - \mathbf{z} \rangle + \frac{\eta}{2} \|\mathcal{P}_\Omega(\mathbf{z} - \mathbf{y})\|_2^2$$

- The ADMM scheme:

$$\begin{cases} \mathbf{x} := \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) & \text{(Nuclear norm minimization)} \\ \mathbf{z} := \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) & \text{(Closed-form solution)} \\ \mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z}) & \text{(Standard update)} \end{cases}$$

- Optimize \mathbf{x} ?

$$\|\mathcal{C}(\mathbf{x})\|_* = \|\mathcal{F}(\mathbf{x})\|_1 \quad \& \quad \frac{1}{2} \|\boldsymbol{\ell} * \mathbf{x}\|_2^2 = \frac{1}{2T} \|\mathcal{F}(\boldsymbol{\ell}) \circ \mathcal{F}(\mathbf{x})\|_2^2$$

Nuclear norm minimization \Rightarrow **ℓ_1 -norm minimization with FFT**

³ $\mathbf{w} \in \mathbb{R}^T$ (Lagrange multiplier); $\langle \cdot, \cdot \rangle$ (inner product).

Laplacian Convolutional Representation

- Optimize \mathbf{x} via FFT (in $\mathcal{O}(T \log T)$ time):

$$\begin{aligned}\mathbf{x} &:= \arg \min_{\mathbf{x}} \|\mathcal{C}(\mathbf{x})\|_* + \frac{\gamma}{2} \|\ell * \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{w}/\lambda\|_2^2 \\ \Rightarrow \hat{\mathbf{x}} &:= \arg \min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1 + \frac{\gamma}{2T} \|\hat{\ell} \circ \hat{\mathbf{x}}\|_2^2 + \frac{\lambda}{2T} \|\hat{\mathbf{x}} - \hat{\mathbf{z}} + \hat{\mathbf{w}}/\lambda\|_2^2\end{aligned}$$

where we introduce $\{\hat{\ell}, \hat{\mathbf{x}}, \hat{\mathbf{z}}, \hat{\mathbf{w}}\} \triangleq \mathcal{F}\{\ell, \mathbf{x}, \mathbf{z}, \mathbf{w}\}$ (i.e., FFT).

ℓ_1 -norm Minimization in Complex Space (Liu & Zhang'23)

For any optimization problem in the form of ℓ_1 -norm minimization in complex space:

$$\min_{\hat{\mathbf{x}}} \|\hat{\mathbf{x}}\|_1 + \frac{\delta}{2} \|\hat{\mathbf{x}} - \hat{\mathbf{h}}\|_2^2$$

with complex-valued $\hat{\mathbf{x}}, \hat{\mathbf{h}} \in \mathbb{C}^T$ and weight parameter δ , element-wise, the solution is given by

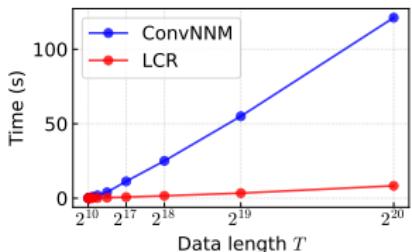
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - 1/\delta\}, t = 1, \dots, T.$$

Laplacian Convolutional Representation

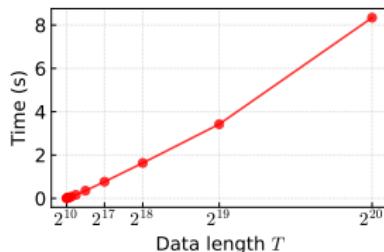
Empirical time complexity

On the synthetic data $\mathbf{y} \in \mathbb{R}^T$ with $T \in \{2^{10}, 2^{11}, \dots, 2^{20}\}$

- Ours: **LCR**
 - An FFT implementation in $\mathcal{O}(T \log T)$
 - The logarithmic factor $\log T$ makes the FFT highly efficient
- Baseline: **ConvNNM⁴** ([Liu'22](#), [Liu & Zhang'23](#))
 - Convolution matrix $C_{\tilde{\tau}}(\mathbf{y}) \in \mathbb{R}^{T \times \tilde{\tau}}$ with kernel size $\tilde{\tau} = 2^4$
 - Singular value thresholding in $\mathcal{O}(\tilde{\tau}^2 T)$



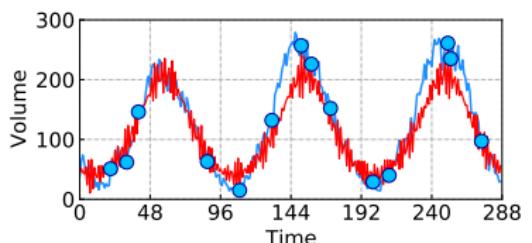
ConvNNM vs. LCR



LCR

⁴Convolution nuclear norm minimization.

Univariate Traffic Time Series Imputation

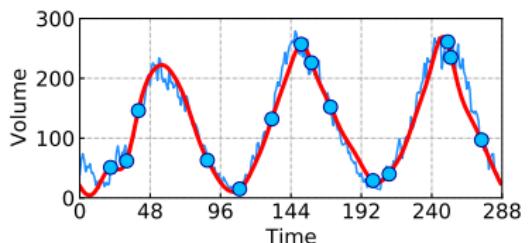


CircNNM:

$$\min_{\mathbf{x}} \underbrace{\|\mathcal{C}(\mathbf{x})\|_*}_{\text{global}}$$

$$\text{s. t. } \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon$$

↓ Plus temporal regularization

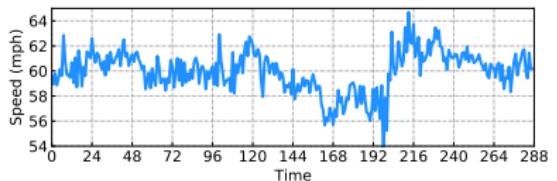


LCR:

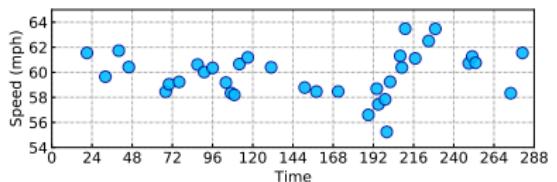
$$\min_{\mathbf{x}} \underbrace{\|\mathcal{C}(\mathbf{x})\|_*}_{\text{global}} + \frac{\gamma}{2} \underbrace{\|\ell * \mathbf{x}\|_2^2}_{\text{local}}$$

$$\text{s. t. } \|\mathcal{P}_\Omega(\mathbf{x} - \mathbf{y})\|_2 \leq \epsilon$$

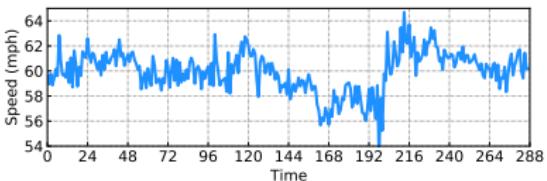
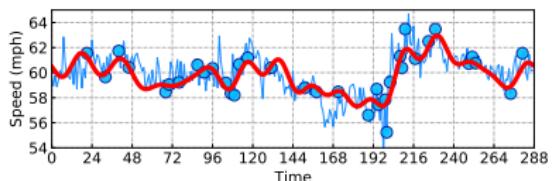
Univariate Traffic Time Series Imputation



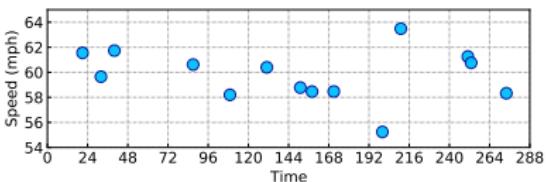
↓ Mask 90% observations



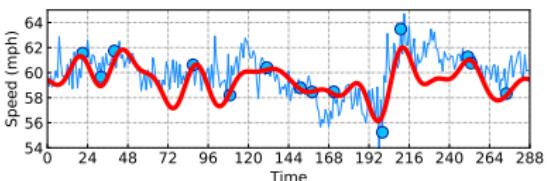
↓ Reconstruct time series



↓ Mask 95% observations



↓ Reconstruct time series



LCR can reconstruct traffic time series from very sparse data.

Large-Scale Traffic Data Imputation

LCR vs. baseline (in MAPE/RMSE)

- PeMS-4W: California freeway traffic speed dataset ($Y \in \mathbb{R}^{11160 \times 8064}$)

Model	Missing rate			
	30%	50%	70%	90%
LCR-2D	1.50/1.49	1.76/1.69	2.07/2.06	3.19/3.05
LCR _N	1.48/1.50	1.73/1.73	2.07/2.12	3.24/3.22
LCR	1.50/1.49	1.76/1.69	2.08/2.07	3.21/3.06
CTNNM	2.26/1.84	2.67/2.14	3.40/2.66	5.22/3.90
CircNNM	2.26/1.84	2.69/2.15	3.43/2.67	5.34/3.96
LRMC	2.04/1.80	2.43/2.12	3.08/2.66	6.05/4.43
HaLRTC	1.98/1.73	2.22/1.98	2.84/2.49	4.39/3.66
LRTC-TNN	1.68/1.55	1.93/1.77	2.33/2.14	3.40/3.10
NoTMF	2.95/2.65	3.05/2.73	3.33/2.97	5.22/4.71

Results

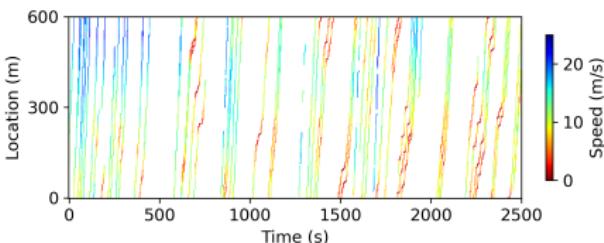
- LCR-2D > CTNNM: The importance of temporal regularization.
- CTNNM \geq CircNNM: Cyclic tensor is superior to circulant matrix.
- LCR > LRMC/LRTC: The importance of global/local modeling.

$\mathcal{O}(NT \log(NT))$ (FFT) vs. $\mathcal{O}(\min\{N^2T, NT^2\})$ (SVD)

Hankel Tensor Factorization

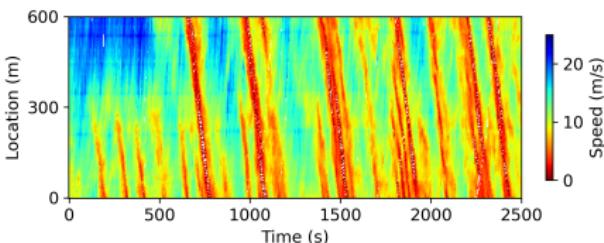
Motivation: Spatiotemporal data reconstruction

- Sparse speed field reconstruction problem in vehicular traffic flow.



200-by-500 matrix
(NGSIM)

Reconstruct speed field from
5% sparse trajectories?



How to characterize both spatial and temporal dependencies?

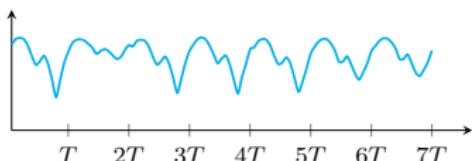
Hankel Tensor Factorization

- Hankel matrix

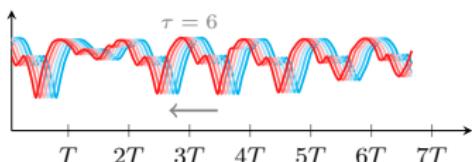
- Given $\mathbf{x} = (1, 2, 3, 4, 5)^\top$ and window length $\tau = 2$, we have

$$\mathcal{H}_\tau(\mathbf{x}) = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \\ 4 & 5 \end{bmatrix} \in \mathbb{R}^{4 \times 2}$$

- Automatic temporal modeling



Traffic time series



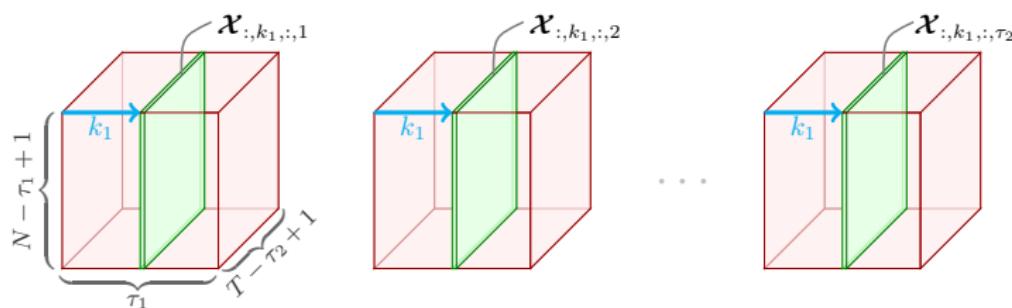
Hankel matrix

Hankel Tensor Factorization

- Hankel tensor: Given any matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$, we have

$$\mathcal{X} \triangleq \mathcal{H}_{\tau_1, \tau_2}(\mathbf{X})$$

- Window lengths: $\tau_1, \tau_2 \in \mathbb{N}^+$;
- Tensor size: $(N - \tau_1 + 1) \times \tau_1 \times (T - \tau_2 + 1) \times \tau_2$;



(Figure) 4th-order Hankel tensor: A sequence of third-order tensors.

- Slice: $\mathcal{X}_{:,k_1,:,:k_2}$, $\forall k_1, k_2$;
- Slice size: $(N - \tau_1 + 1) \times (T - \tau_2 + 1)$.

Hankel Tensor Factorization

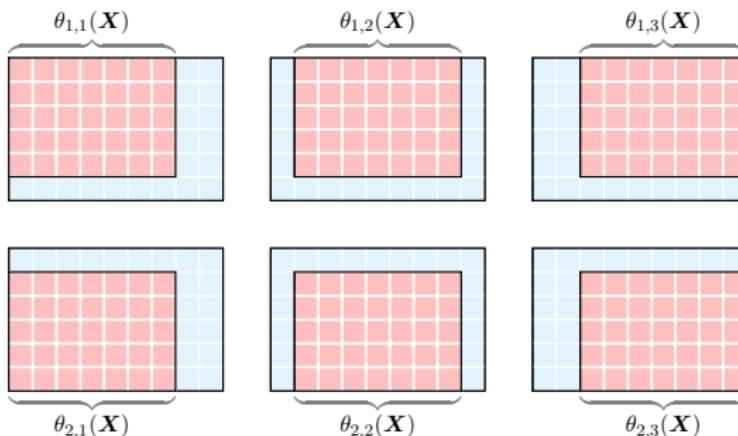
Hankel indexing

- Sampling function for the Hankel tensor:

$$\theta_{k_1, k_2}(\mathbf{X}) \triangleq [\mathcal{H}_{\tau_1, \tau_2}(\mathbf{X})]_{:, k_1, :, k_2},$$

referring to as the tensor slice with $k_1 \in \{1, \dots, \tau_1\}$, $k_2 \in \{1, \dots, \tau_2\}$.

- [Importance] Developing memory-efficient algorithms



- Tensor slices $\theta_{k_1, k_2}(\mathbf{X})$ vs. data matrix \mathbf{X}

Hankel Tensor Factorization

Ours:

- Convolutional tensor decomposition (circular convolution \star_{row}):

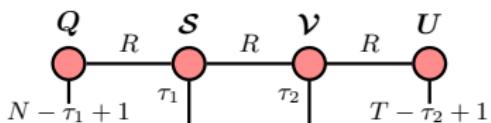
$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \star_{\text{row}} \mathbf{s}_{k_1}^{\top})(\mathbf{U} \star_{\text{row}} \mathbf{v}_{k_2}^{\top})^{\top}$$

Baselines:

- Tensor-train decomposition:

$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \mathbf{S}_{k_1})(\mathbf{U} \mathbf{V}_{k_2})^{\top}$$

- $\{\mathbf{S}_{k_1}, \mathbf{V}_{k_2}\}$ are **circulant matrices** \Rightarrow convolutional decomposition
- $\{\mathbf{S}_{k_1}, \mathbf{V}_{k_2}\}$ are **diagonal matrices** \Rightarrow CP decomposition



- CP tensor decomposition (Khatri-Rao product \odot):

$$\theta_{k_1, k_2}(\mathbf{Y}) \approx (\mathbf{Q} \odot \mathbf{s}_{k_1}^{\top})(\mathbf{U} \odot \mathbf{v}_{k_2}^{\top})^{\top}$$

Hankel Tensor Factorization

HTF (convolutional decomposition)

- Optimization problem:

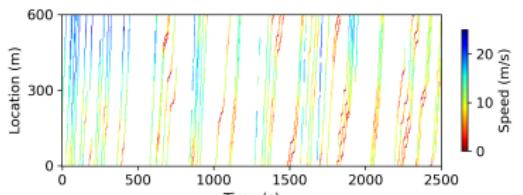
$$\min_{Q, S, U, V} \frac{1}{2} \underbrace{\sum_{k_1, k_2} \left\| \mathcal{P}_{\Omega_{k_1, k_2}} (\theta_{k_1, k_2}(\mathbf{Y}) - (Q \star_{\text{row}} s_{k_1}^{\top})(U \star_{\text{row}} v_{k_2}^{\top})^{\top}) \right\|_F^2}_{\text{Tensor decomposition on Hankel tensor slices}} + \frac{\rho}{2} (\|Q\|_F^2 + \|S\|_F^2 + \|U\|_F^2 + \|V\|_F^2)$$

- Alternating minimization (let f be the obj.):

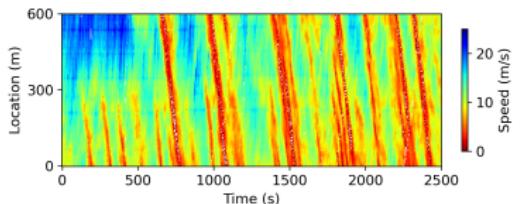
$$\begin{cases} Q := \{Q \mid \frac{\partial f}{\partial Q} = \mathbf{0}\} & \text{(conjugate gradient)} \\ s_{k_1} := \{s_{k_1} \mid \frac{\partial f}{\partial s_{k_1}} = \mathbf{0}\}, \forall k_1 & \text{(conjugate gradient)} \\ U := \{U \mid \frac{\partial f}{\partial U} = \mathbf{0}\} & \text{(conjugate gradient)} \\ v_{k_2} := \{v_{k_2} \mid \frac{\partial f}{\partial v_{k_2}} = \mathbf{0}\}, \forall k_2 & \text{(conjugate gradient)} \end{cases}$$

- Memory-efficient but still computationally costly!

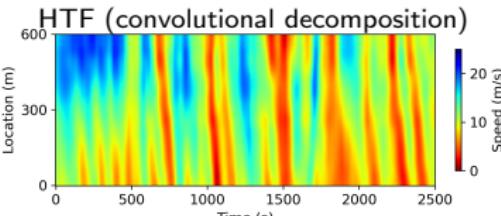
Extreme Missing Traffic Data Imputation



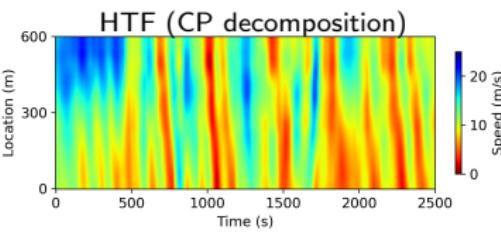
Sparse speed field



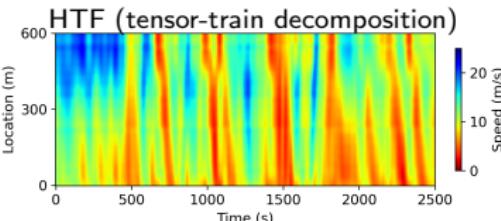
Ground truth speed field



MAPE = 51.92%



MAPE = 53.93%



MAPE = 56.48%

Extreme Missing Traffic Data Imputation

HTF vs. baseline (in MAPE/RMSE)

- On the Seattle freeway traffic speed dataset ($Y \in \mathbb{R}^{323 \times 8064}$)

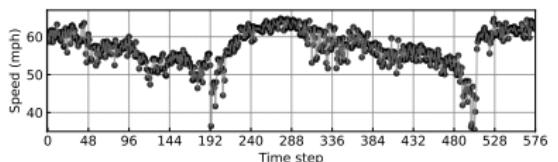
Model	Random missing rate			
	80%	85%	90%	95%
HTF (convolution)	6.21/3.88	6.51/4.06	6.98/4.30	8.02/4.84
HTF (tensor-train)	8.75/5.16	9.86/5.76	9.24/5.36	9.89/5.70
HTF (CP)	7.25/4.33	7.93/4.66	8.61/4.96	9.25/5.20
LATC	6.50/4.00	6.90/4.21	7.47/4.51	8.75/5.05
LRTC-TNN	6.97/4.24	7.43/4.43	8.19/4.81	9.60/5.55
LCR-2D	6.75/4.15	7.31/4.38	7.96/4.71	9.78/5.39
BTMF	6.85/4.17	7.36/4.42	8.13/4.79	9.63/5.48

Results

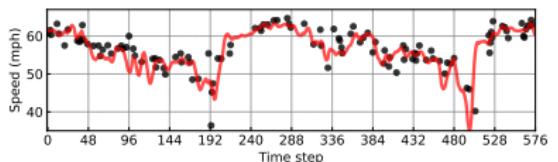
- Convolutional tensor decomposition outperforms both tensor-train and CP tensor decomposition.
- Our HTF model performs better than state-of-the-art baseline models.

Extreme Missing Traffic Data Imputation

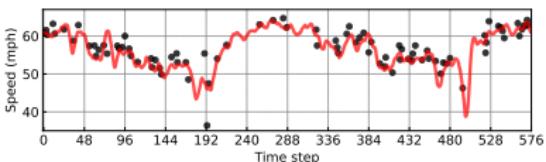
- Example: 1st time series within two days.



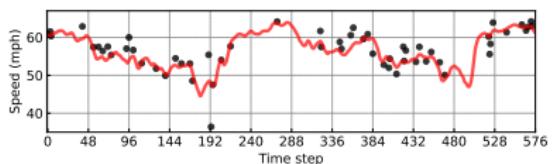
Original time series



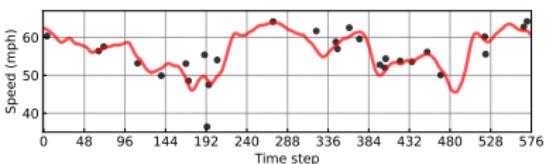
80% missing rate



85% missing rate



90% missing rate



95% missing rate

Conclusion

Sparse	
High-dimensional	
Multidimensional	
Noises & outliers	
Nonstationary	

Low-rank framework:

- NoTMF: matrix factorization
- LCR: circulant matrix nuclear norm minimization
- HTF: tensor factorization

⇒ Temporal modeling:

- NoTMF: seasonal differenced vector autoregression
- LCR: temporal smoothing
- HTF: automatic temporal modeling with Hankel tensor

Highlights & Contributions

Matrix nuclear norm
($\|\mathbf{X}\|_*$) minimization

Candès & Recht'09

Singular value
thresholding

Cai et al.'10

Truncated nuclear
norm ($\|\mathbf{X}\|_{r,*}$, $r \in \mathbb{N}^+$)
minimization

Zhang et al.'12
Hu et al.'12

Tensor nuclear norm
($\|\mathcal{X}\|_*$) minimization

Liu et al.'13

Circulant/Convolution
nuclear norm
($\|\mathcal{C}(\mathbf{X})\|_*$ or $\|\mathcal{C}_{\tilde{\tau}}(\mathbf{X})\|_*$)
minimization

Liu'22
Liu & Zhang'23

Low-rank Hankel
matrix/tensor
($\mathcal{H}_\tau(\cdot)$) completion

Yokota et al.'18
Sedighin et al.'20
Cai et al.'21
Yamamoto et al.'22

Tensor nuclear norm
minimization with
linear transform

Lu et al.'19

Generalized nonconvex
nonsmooth low-rank
minimization

Lu et al.'14

(Ours) LCR:

- ✓ Local trend modeling
- ✓ An FFT implementation

(Ours) HTF:

- ✓ Memory-efficient
- ✓ Conv. para.

References

A short list:

- ([Candès & Recht'09](#)) "Exact matrix completion via convex optimization." *Foundations of Computational Mathematics*. 2009, 9(6): 717-772.
- ([Cai et al.'10](#)) "A singular value thresholding algorithm for matrix completion." *SIAM Journal on optimization*. 2010, 20(4): 1956-1982.
- ([Zhang et al.'12](#)) "Matrix completion by truncated nuclear norm regularization." *IEEE Conference on computer vision and pattern recognition*. 2012.
- ([Hu et al.'12](#)) "Fast and accurate matrix completion via truncated nuclear norm regularization." *IEEE transactions on pattern analysis and machine intelligence*. 2012, 35(9): 2117-2130.
- ([Lu et al.'14](#)) "Generalized nonconvex nonsmooth low-rank minimization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
- ([Gultekin & Paisley'18](#)) "Online forecasting matrix factorization." *IEEE Transactions on Signal Processing*. 2018, 67(5): 1223-1236.
- ([Yokota et al.'18](#)) "Missing slice recovery for tensors using a low-rank model in embedded space." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- ([Lu et al.'19](#)) "Tensor robust principal component analysis with a new tensor nuclear norm." *IEEE transactions on pattern analysis and machine intelligence*. 2019, 42(4): 925-938.
- ([Cai et al.'21](#)) "Accelerated structured alternating projections for robust spectrally sparse signal recovery." *IEEE Transactions on Signal Processing*. 2021, 69: 809-821.
- ([Chen & Sun'22](#)) "Bayesian temporal factorization for multidimensional time series prediction." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2022, 44(9): 4659-4673.
- ([Liu'22](#)) "Time series forecasting via learning convolutionally low-rank models." *IEEE Transactions on Information Theory*. 2022, 68(5): 3362-3380.
- ([Liu & Zhang'23](#)) "Recovery of future data via convolution nuclear norm minimization." *IEEE Transactions on Information Theory*. 2023, 69(1): 650-665.



POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉnierie



Thanks for your attention!

Any Questions?

Slides: <https://xinychen.github.io/slides/sustech23.pdf>

About me:

- Homepage: <https://xinychen.github.io>
- How to reach me: chenxy346@gmail.com

Research Interests

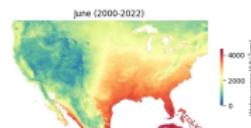
- Machine learning & spatiotemporal data modeling



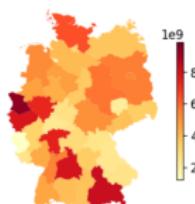
Transportation



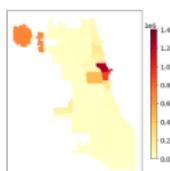
Mobile service



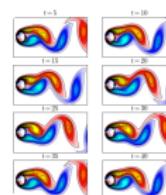
Climate



Energy



Mobility



Dynamical system

- Urban science
- Intelligent transportation systems

Past Works

Spatiotemporal traffic data imputation:

1. Xinyu Chen, Zhaocheng He, Jiawei Wang (2018). Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition. *Transportation Research Part C: Emerging Technologies*. 86: 59-77. (100+ citations)
2. Xinyu Chen, Zhaocheng He, Lijun Sun (2019). A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*. 98: 73-84. (200+ citations, ESI highly cited paper)
3. Xinyu Chen, Zhaocheng He, Yixian Chen, Yuhuan Lu, Jiawei Wang (2019). Missing traffic data imputation and pattern discovery with a Bayesian augmented tensor factorization model. *Transportation Research Part C: Emerging Technologies*. 104: 66-77. (100+ citations)
4. Xinyu Chen, 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. (100+ citations)
5. Xinyu Chen, Yixian Chen, Nicolas Saunier, Lijun Sun (2021). Scalable low-rank tensor learning for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*. 129: 103226.

Past Works

Spatiotemporal traffic data imputation:

6. Xinyu Chen, Mengying Lei, Nicolas Saunier, Lijun Sun (2022). Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems*. 23 (8): 12301-12310. (50+ citations, ESI hot paper)

Spatiotemporal time series forecasting:

7. Xinyu Chen, Lijun Sun (2022). Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 44 (9): 4659-4673. (150+ citations, ESI hot paper & ESI highly cited paper)

Spatiotemporal pattern discovery:

8. Xinyu Chen, Chengyuan Zhang, Xiaoxu Chen, Nicolas Saunier, Lijun Sun (2023). Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. *IEEE Transactions on Knowledge and Data Engineering*. Early access.

Past Works

A strong advocate of open-source and reproducible research:

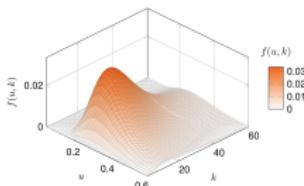
- <https://github.com/xinyuchen>

Algorithms



transdim
(1.1k stars)

Tools



awesome-latex-drawing
(1.2k stars)

Tutorials



latex-cookbook
(1.1k stars)
(THU Press)

Future Plan

Broad interests:

- Data Science
- Machine Learning
- AI for Science

Research directions:

- Urban science
- Human mobility modeling
- Geospatial data analysis
- Intelligent & sustainable urban systems
- Optimization & decision making

Goals: Solving many scientific, mathematical, and engineering problems with AI algorithms.

Website: <https://spatiotemporal-data.github.io>