POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

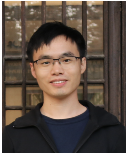
IVADO

Low-Rank Matrix and Tensor Methods for Spatiotemporal Traffic Data Modeling

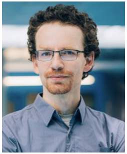
Xinyu Chen

University of Montreal, Canada

May 11, 2023



Ph.D. Candidate
Xinyu Chen



Supervisor
Prof. Nicolas Saunier

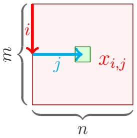
Co-supervisor
Prof. Lijun Sun

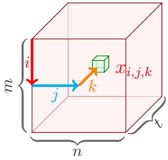
Outline

- **Spatiotemporal Traffic Data**
- **Spatiotemporal Traffic Data Imputation**
  Laplacian convolutional representation
  Hankel tensor factorization
- **Sparse Traffic Flow Forecasting**
- **Dynamic Pattern Discovery**
- **Conclusion**

Matrix & Tensor

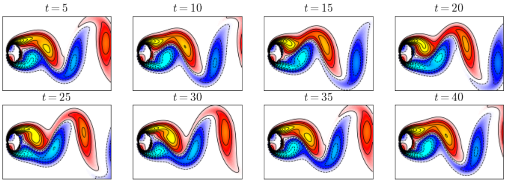
- What is tensor? $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ vs. $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m \times n \times t}$





- Tensors are everywhere!

Color image with RGB channels

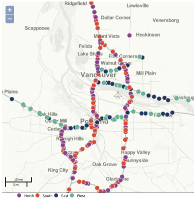


Dynamical system (fluid flow)

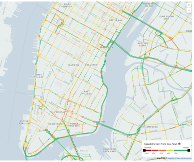
Spatiotemporal Traffic Data

- Spatiotemporal traffic data are indeed matrices or tensors.

Network



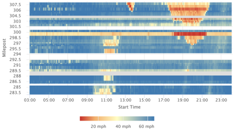
Highway network



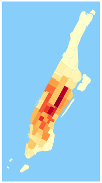
Urban road network

$\Longrightarrow$

Data & Representation

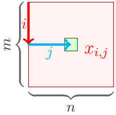


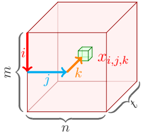
Speed field



OD matrices over time

Matrix **Tensor**





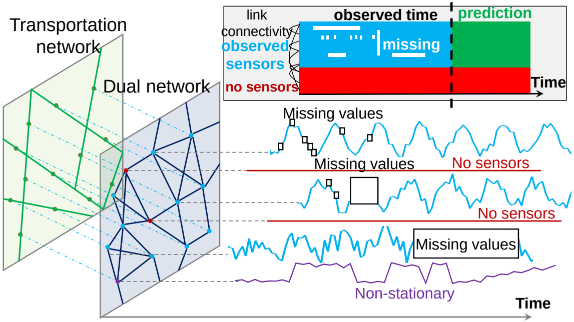
Spatiotemporal Traffic Data Imputation

❶ X. Chen, Z. Cheng, N. Saunier, L. Sun (2022). Laplacian convolutional representation for traffic time series imputation. arXiv preprint arXiv:2212.01529.

(Under 1st review at IEEE Transactions on Signal Processing)

❷ X. Chen Z. Cheng, L. Sun, N. Saunier (2023). Memory-efficient Hankel tensor factorization for extreme missing traffic data imputation. (coming soon)

Laplacian Convolutional Representation



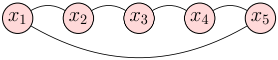
Motivation:

- How to characterize both global and local trends in sparse traffic data?

Laplacian Convolutional Representation

Reformulate Laplacian regularization with circular convolution.

- Intuition of (circulant) Laplacian matrix.



Undirected and circulant graph

$\xrightarrow{\text{Modeling}}$

$$\boldsymbol{L}=\begin{bmatrix}2 & -1 & 0 & 0 & -1\\ -1 & 2 & -1 & 0 & 0\\ 0 & -1 & 2 & -1 & 0\\ 0 & 0 & -1 & 2 & -1\\ -1 & 0 & 0 & -1 & 2\end{bmatrix}$$

(Circulant) Laplacian matrix

- Laplacian kernel: $\boldsymbol{\ell}=(2,-1,0,0,-1)^\top$.
- Define Laplacian kernel:

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

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

- Local trend modeling via (Laplacian) temporal regularization:

$$\mathcal{R}_\tau(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{L}\boldsymbol{x}\|_2^2=\frac{1}{2}\|\boldsymbol{\ell}\star\boldsymbol{x}\|_2^2=\frac{1}{2T}\|\mathcal{F}(\boldsymbol{\ell})\circ\mathcal{F}(\boldsymbol{x})\|_2^2$$

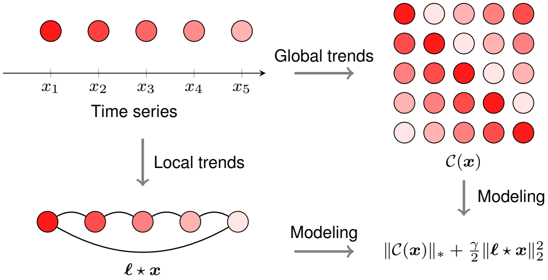
Laplacian Convolutional Representation

Laplacian Convolutional Representation (LCR)

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

$$\begin{aligned} \min_{\boldsymbol{x}} \; & \|\mathcal{C}(\boldsymbol{x})\|_* + \gamma \cdot \mathcal{R}_\tau(\boldsymbol{x}) \\ \text{s.t.} \; & \|\mathcal{P}_\Omega(\boldsymbol{x} - \boldsymbol{y})\|_2 \leq \epsilon \end{aligned}$$

where $\mathcal{C}: \mathbb{R}^T \to \mathbb{R}^{T \times T}$ denotes the circulant operator. $\|\cdot\|_*$ denotes the nuclear norm of matrix, namely, the sum of singular values.



Laplacian Convolutional Representation

- LCR model:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \|\mathcal{C}(\boldsymbol{x})\|_* + \gamma \cdot \mathcal{R}_\tau(\boldsymbol{x}) \\ \text{s.t. } & \|\mathcal{P}_\Omega(\boldsymbol{x} - \boldsymbol{y})\|_2 \leq \epsilon \end{aligned}$$

- Augmented Lagrangian function:

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

  where $\boldsymbol{w} \in \mathbb{R}^T$ is the Lagrange multiplier, and $\langle \cdot, \cdot \rangle$ denotes the inner product.

- The ADMM scheme:

$$\begin{cases} \boldsymbol{x} :=& \arg\min_{\boldsymbol{x}} \ \mathcal{L}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}) \\ \boldsymbol{z} :=& \arg\min_{\boldsymbol{z}} \ \mathcal{L}(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{w}) \\ &= \frac{1}{\lambda + \eta}\mathcal{P}_\Omega(\lambda \boldsymbol{x} + \boldsymbol{w} + \eta \boldsymbol{y}) + \frac{1}{\lambda}\mathcal{P}_\Omega^\perp(\lambda \boldsymbol{x} + \boldsymbol{w}) \\ \boldsymbol{w} :=& \boldsymbol{w} + \lambda(\boldsymbol{x} - \boldsymbol{z}) \end{cases}$$

Laplacian Convolutional Representation

- Optimize $\boldsymbol{x}$ via fast Fourier transform (FFT in $\mathcal{O}(T\log T)$ time):

$$\begin{aligned}
\boldsymbol{x} &:= \arg\min_{\boldsymbol{x}}\ \|\mathcal{C}(\boldsymbol{x})\|_* + \frac{\gamma}{2}\|\boldsymbol{\ell}\star\boldsymbol{x}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{x}-\boldsymbol{z}+\boldsymbol{w}/\lambda\|_2^2 \\
\Longrightarrow \hat{\boldsymbol{x}} &:= \arg\min_{\hat{\boldsymbol{x}}}\ \|\hat{\boldsymbol{x}}\|_1 + \frac{\gamma}{2T}\|\hat{\boldsymbol{\ell}}\circ\hat{\boldsymbol{x}}\|_2^2 + \frac{\lambda}{2T}\|\hat{\boldsymbol{x}}-\hat{\boldsymbol{z}}+\hat{\boldsymbol{w}}/\lambda\|_2^2 \\
&= \arg\min_{\hat{\boldsymbol{x}}}\ \|\hat{\boldsymbol{x}}\|_1 + \frac{\lambda}{2T}\|\hat{\boldsymbol{x}}-(\lambda\hat{\boldsymbol{z}}-\hat{\boldsymbol{w}})\oslash(\lambda\hat{\boldsymbol{\ell}}^*\circ\hat{\boldsymbol{\ell}}+\lambda\mathbb{1}_T)\|_2^2
\end{aligned}$$

where $\{\hat{\boldsymbol{\ell}},\hat{\boldsymbol{x}},\hat{\boldsymbol{z}},\hat{\boldsymbol{w}}\}$ refers to $\{\boldsymbol{\ell},\boldsymbol{x},\boldsymbol{z},\boldsymbol{w}\}$ in the frequency domain.

ℓ_1-norm Minimization in Complex Space (Liu & Zhang'22)

For any ℓ_1-norm minimization problem in complex space:

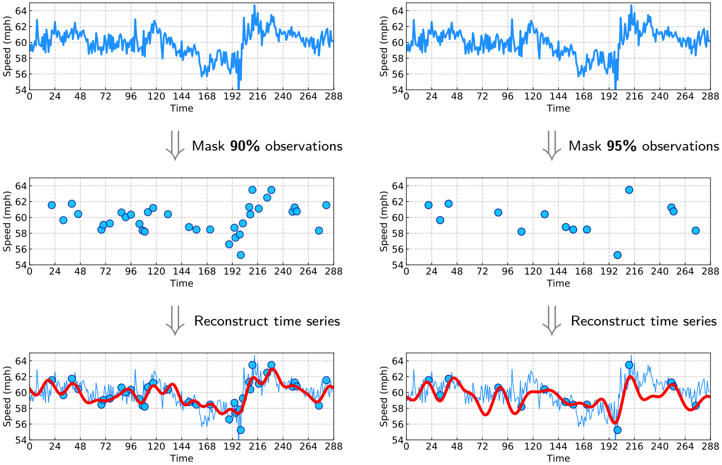
$$\min_{\hat{\boldsymbol{x}}}\ \|\hat{\boldsymbol{x}}\|_1 + \frac{\lambda}{2T}\|\hat{\boldsymbol{x}}-\hat{\boldsymbol{h}}\|_2^2$$

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

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

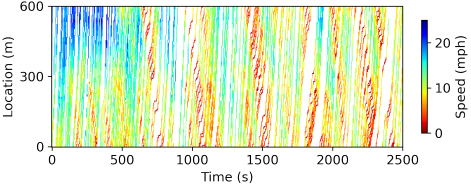
Laplacian Convolutional Representation

- On traffic speed time series:

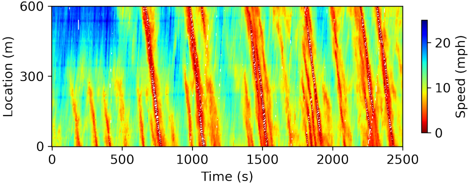


- LCR can characterize both global and local trends and produce accurate results.

Hankel Tensor Factorization



200-by-500 matrix (NGSIM) $\Downarrow$ Reconstruct speed field from 20% sparse trajectories?



Motivation:

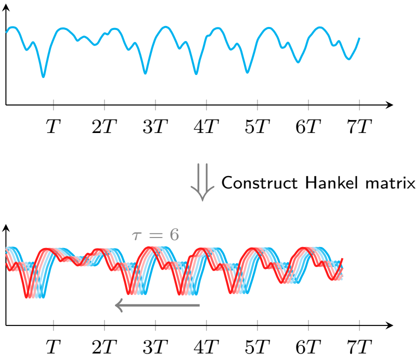
- How to learn from sparse spatiotemporal data?
- How to characterize spatial/temporal local dependencies?

Hankel Tensor Factorization

- Hankel matrix
  - Given $\boldsymbol{y} = (1, 2, 3, 4, 5)^\top$ and window length $\tau = 2$, we have

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

  - Hankel matrix $\mathcal{H}_\tau(\boldsymbol{y})$ on time series $\boldsymbol{y}$:



Hankel Tensor Factorization

- Hankel matrix
  - Given $\boldsymbol{y} = (1, 2, 3, 4, 5)^\top$ and window length $\tau = 2$, we have

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

  - On time series $\boldsymbol{y} = (y_1, y_2, \ldots, y_5)^\top$ with $\tau = 2$:

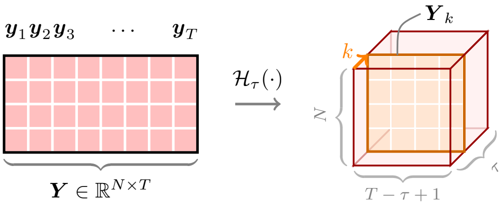
$$\mathcal{H}_\tau(\boldsymbol{y}) = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \\ y_3 & y_4 \\ y_4 & y_5 \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\Longrightarrow \quad \hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \\ \hat{y}_5 \end{bmatrix} = \mathcal{H}_\tau^{-1}\left(\begin{bmatrix} v_1x_1 & v_1x_2 \\ v_2x_1 & v_2x_2 \\ v_3x_1 & v_3x_2 \\ v_4x_1 & v_4x_2 \end{bmatrix}\right) = \begin{bmatrix} v_1x_1 \\ (v_1x_2 + v_2x_1)/2 \\ (v_2x_2 + v_3x_1)/2 \\ (v_3x_2 + v_4x_1)/2 \\ v_4x_2 \end{bmatrix}$$

  - Automatic **temporal** modeling

Hankel Tensor Factorization

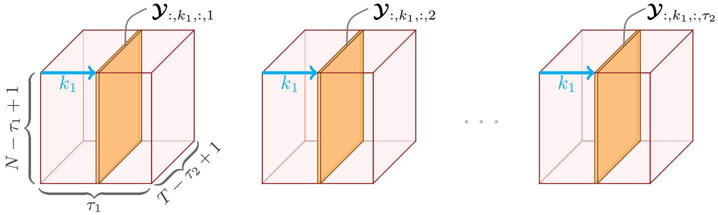
- Hankelization from $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ to Hankel tensor $\mathcal{H}_\tau(\boldsymbol{Y})$.
  - Tensor size: $N \times (T-\tau+1) \times \tau$;
  - Slices: $\boldsymbol{Y}_k = \begin{bmatrix} | & | & & | \\ \boldsymbol{y}_k & \boldsymbol{y}_{k+1} & \cdots & \boldsymbol{y}_{T-\tau+k} \\ | & | & & | \end{bmatrix}, k = 1, 2, \ldots, \tau;$
  - Slice size: $N \times (T-\tau+1)$.



  - Automatic **temporal** modeling

Hankel Tensor Factorization

- Hankelization from $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ to $\boldsymbol{\mathcal{Y}} \triangleq \mathcal{H}_{\tau_1,\tau_2}(\boldsymbol{Y})$ (Hankel tensor).
  - Tensor size: $(N-\tau_1+1) \times \tau_1 \times (T-\tau_2+1) \times \tau_2$;
  - Slice: $\boldsymbol{\mathcal{Y}}_{:,k_1,:,k_2}, \forall k_1, k_2$;
  - Slice size: $(N-\tau_1+1) \times (T-\tau_2+1)$.



  - Automatic **spatial** and **temporal** modeling

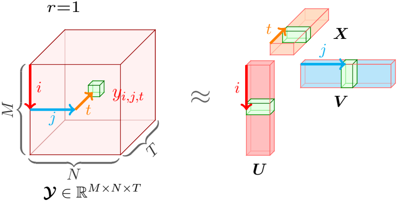
- Revisit tensor factorization (TF)

1927	1960s	1970	2009	2011
Higher-Order SVD	Tucker Decomposition	CP Decomposition	Tensor Decompositions and Applications	Tensor-Train Decomposition
	Ledyard R. Tucker	J. Douglas Carroll Jih-Jie Chang Richard A. Harshman		
Frank Lauren Hitchcock			Tamara G. Kolda	Ivan Oseledets

- Revisit tensor factorization (TF)

1927	1960s	1970	2009	2011
Higher-Order SVD	Tucker Decomposition	CP Decomposition	Tensor Decompositions and Applications	Tensor-Train Decomposition
 Frank Lauren Hitchcock	Ledyard R. Tucker	J. Douglas Carroll Jih-Jie Chang Richard A. Harshman	 Tamara G. Kolda	 Ivan Oseledets

- **CP decomposition**: Factorize $\boldsymbol{\mathcal{Y}}$ into the combination of rank-R factor matrices, i.e., $\boldsymbol{\mathcal{Y}} \approx \sum_{r=1}^{R} \boldsymbol{u}_r \otimes \boldsymbol{v}_r \otimes \boldsymbol{x}_r$.



Hankel Tensor Factorization

- Hankel tensor factorization:

$$\min_{\boldsymbol{Q},\boldsymbol{S},\boldsymbol{U},\boldsymbol{V}} \ \frac{1}{2}\left\|\mathcal{P}_{\dot{\Omega}}\Big(\mathcal{H}_{\tau_1,\tau_2}(\boldsymbol{Y})-\sum_{r=1}^{R}\boldsymbol{q}_r\otimes\boldsymbol{s}_r\otimes\boldsymbol{u}_r\otimes\boldsymbol{v}_r\Big)\right\|_F^2$$

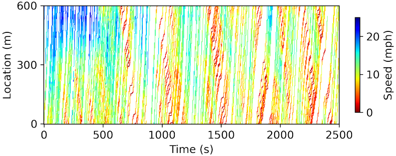
- Advantage/disadvantage:
  - ✓ Automatic spatial and temporal modeling
  - ✗ High memory consumption
- Space complexity:

$$\mathcal{O}(\tau_1\tau_2(N-\tau_1+1)(T-\tau_2+1))$$

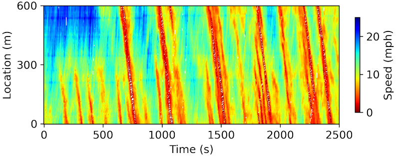
- **(Contribution)** Reduce the space complexity to $\mathcal{O}(NT)$.

Hankel Tensor Factorization

Which Model Is Better?

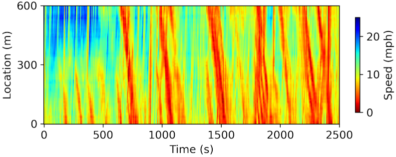


Sparse speed field

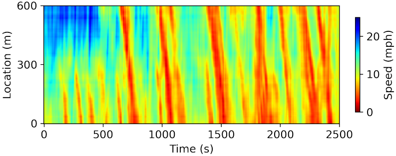


Ground truth speed field

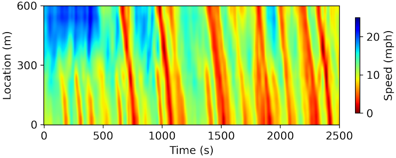
Matrix factorization



Smoothing matrix factorization

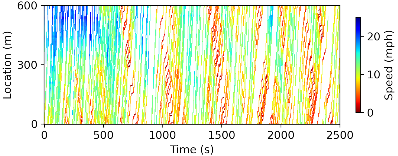


Hankel tensor factorization

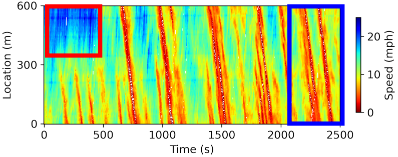


Hankel Tensor Factorization

Which Model Is Better?

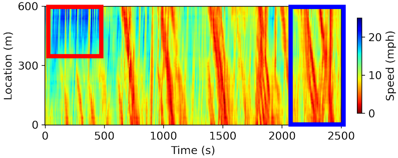


Sparse speed field

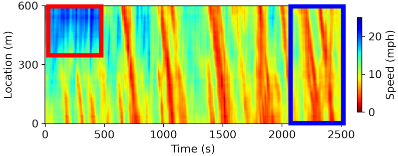


Ground truth speed field

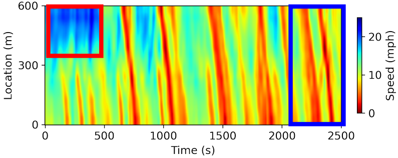
Matrix factorization



Smoothing matrix factorization



Hankel tensor factorization



Sparse Traffic Flow Forecasting

❸ X. Chen, L. Sun (2022). Bayesian temporal factorization for multidimensional time series prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (9): 4659–4673.

- 100+ citations on Google Scholar
- ESI highly cited paper (top 1%)
- ESI hot paper (top 0.1%)

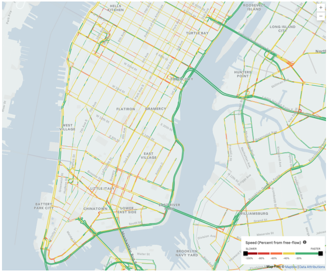
❹ X. Chen, C. Zhang, X.-L. Zhao, N. Saunier, L. Sun (2022). Nonstationary temporal matrix factorization for sparse traffic time series forecasting. arXiv preprint arXiv:2203.10651.

(Under 2nd review at Transportation Research Part C: Emerging Technologies)

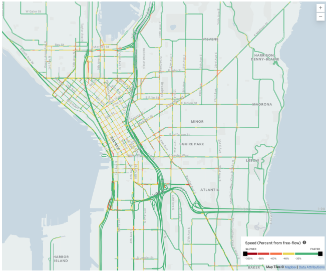
Sparse Traffic Flow Forecasting

Motivation:

- Uber (hourly) movement speed data[1]



NYC movement



Seattle movement

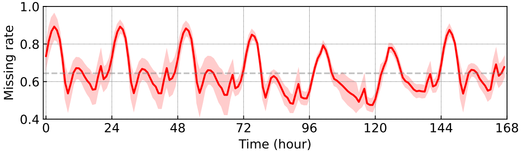
- The average speed on a given road segment for each hour of each day.
- Hourly speeds are computed when road segments have 5+ unique trips.
- **Issue**: insufficient sampling of ridesharing vehicles on the road network.

[1] https://movement.uber.com/

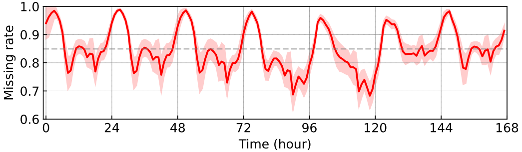
Sparse Traffic Flow Forecasting

High-dimensionality & Sparsity

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

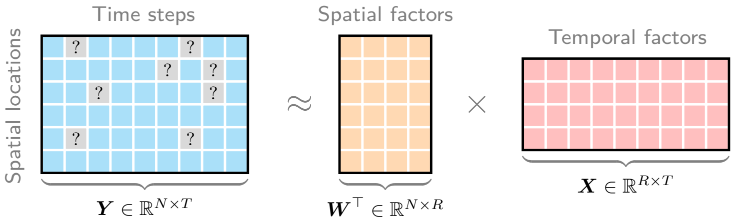


- **Seattle** movement speed data (2019)
  - **63,490** road segments & 8,760 time steps (hours)
  - Whole missing rate: **84.95%**



Sparse Traffic Flow Forecasting

- Spatiotemporal data can be reconstructed by low-dimensional latent factors!



- MF optimization problem

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

with factor matrices $\boldsymbol{W}$ and $\boldsymbol{X}$.

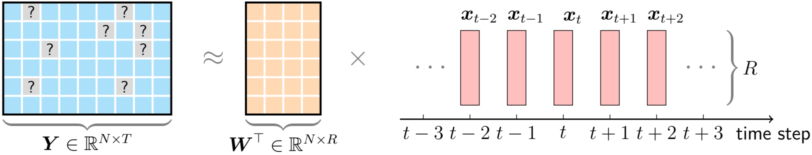
- Disadvantages:
  - Cannot capture temporal correlations.
  - Cannot perform time series forecasting.

Sparse Traffic Flow Forecasting

Temporal matrix factorization (Yu et al.'16; Chen & Sun'22)

Given any partially observed time series data $\boldsymbol{Y} \in \mathbb{R}^{N \times T}$ with observed index set Ω, then temporal matrix factorization assumes a dth-order vector autoregressive (VAR) process on the temporal factor matrix:

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



GitHub repositories:

- **transdim**: Machine learning for spatiotemporal traffic data imputation and forecasting. (960+ stars & 270+ forks)

  https://github.com/xinychen/transdim
- **tracebase**: Multivariate time series forecasting on high-dimensional and sparse Uber movement speed data. (30+ stars)

  https://github.com/xinychen/tracebase
- **awesome-latex-drawing**: Academic drawing examples in LaTeX. (1,000+ stars & 140+ forks)

  https://github.com/xinychen/awesome-latex-drawing

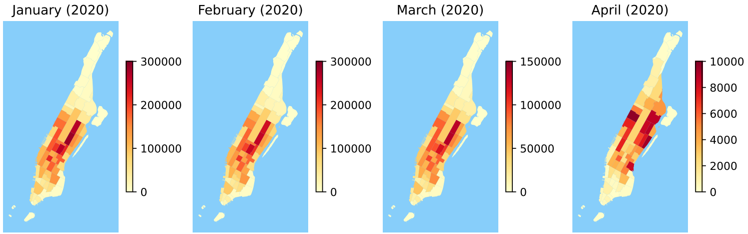
Dynamic Pattern Discovery

❺ X. Chen, C. Zhang†, X. Chen, N. Saunier, L. Sun (2022). Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. arXiv preprint arXiv:2211.15482.

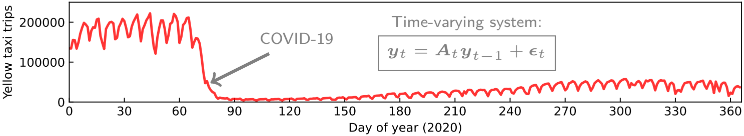
(Under 2nd review at IEEE Transactions on Knowledge and Data Engineering)

Motivation:

- NYC (yellow) taxi data[2]



5,455,923 pickup trips | 5,213,247 pickup trips | 2,752,684 pickup trips | 192,024 pickup trips



- How to characterize the dynamic patterns?

[2] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Dynamic Pattern Discovery

- Given a sequence of spatiotemporal measurements $\boldsymbol{y}_t \in \mathbb{R}^N,\ t = 1, 2, \ldots, T$

$$\min_{\{\boldsymbol{A}_t\}} \underbrace{\frac{1}{2}\sum_t \left\|\boldsymbol{y}_t - \boldsymbol{A}_t \boldsymbol{y}_{t-1}\right\|_2^2}_{\text{Time-varying autoregression}}$$

[**Over-parameterization**] $\mathcal{O}(N^2(T-1))$ parameters vs. $\mathcal{O}(NT)$ data.

- **(Ours)** Parameterize coefficients via TF:

$$\min_{\boldsymbol{W},\boldsymbol{G},\boldsymbol{V},\boldsymbol{X}} \underbrace{\frac{1}{2}\sum_t \left\|\boldsymbol{y}_t - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t^\top \otimes \boldsymbol{V})^\top \boldsymbol{y}_{t-1}\right\|_2^2}_{\text{Let } \boldsymbol{A}_t = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{W} \times_2 \boldsymbol{V} \times_3 \boldsymbol{x}_t^\top \text{ be the TF}}$$

- Alternating minimization (Let f be the obj.)

$$\boldsymbol{W} := \{\boldsymbol{W} \mid \frac{\partial f}{\partial \boldsymbol{W}} = \boldsymbol{0}\} \qquad \boldsymbol{G} := \{\boldsymbol{G} \mid \frac{\partial f}{\partial \boldsymbol{G}} = \boldsymbol{0}\}$$

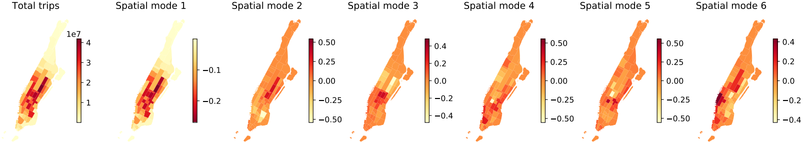
$$\boldsymbol{V} := \{\boldsymbol{V} \mid \frac{\partial f}{\partial \boldsymbol{V}} = \boldsymbol{0}\} \qquad \boldsymbol{x}_t := \{\boldsymbol{x}_t \mid \frac{\partial f}{\partial \boldsymbol{x}_t} = \boldsymbol{0}\}$$

- Solve each subproblem by **conjugate gradient** and **least squares**.

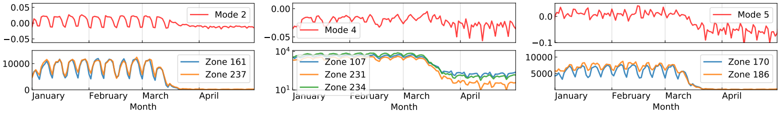
- Time-varying autoregression with TF

$$\min_{\boldsymbol{W},\boldsymbol{G},\boldsymbol{V},\boldsymbol{X}} \ \frac{1}{2}\sum_t \left\| \boldsymbol{y}_t - \boldsymbol{W}\boldsymbol{G}(\boldsymbol{x}_t^\top \otimes \boldsymbol{V})^\top \boldsymbol{y}_{t-1} \right\|_2^2$$

- **NYC taxi dataset** (pickup)



Spatial modes in $\boldsymbol{W}$



Mode #2 & taxi trips (2020) Mode #4 & taxi trips (2020) Mode #5 & taxi trips (2020)

Temporal modes in $\boldsymbol{X}$

- Produce interpretable patterns and identify the changing point of system (mainly due to COVID-19).

Conclusion

Prior Works

Other studies about spatiotemporal data imputation:

❻ X. Chen, M. Lei, N. Saunier, L. Sun (2022). Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. IEEE Transactions on Intelligent Transportation Systems, 23 (8): 12301–12310.

❼ X. Chen, Y. Chen, N. Saunier, L. Sun (2021). Scalable low-rank tensor learning for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies, 129: 103226.

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

❾ X. Chen, Z. He, Y. Chen, Y. Lu, J. 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.

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

References

A short list:

- **[Liu & Zhang'22]** G. Liu and W. Zhang (2022). Recovery of future data via convolution nuclear norm minimization. IEEE Transactions on Information Theory, 69(1), 650–665.
- **[Yu et al.'16]** H.-F. Yu, N. Rao, and I. S. Dhillon (2016). Temporal regularized matrix factorization for high-dimensional time series prediction. Advances in neural information processing systems (NIPS).

Thanks for your attention!

Any Questions?

Slides: https://xinychen.github.io/slides/traffic_data_modeling.pdf

About me:

- Homepage: https://xinychen.github.io
- Google Scholar: user=mCrW04wAAAAJhl (**690 citations**)
- GitHub: https://github.com/xinychen (**3.2k+ stars**)
- Blog: https://medium.com/@xinyu.chen (**70k+ views**)
- How to reach me: chenxy346@gmail.com