POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

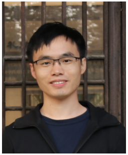
IVADO

Low-Rank Matrix and Tensor Methods for Spatiotemporal Data Modeling

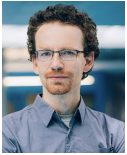
Xinyu Chen

University of Montreal, Canada

April 21, 2023



Ph.D. Candidate
Xinyu Chen



Supervisor
Prof. Nicolas Saunier

Co-supervisor
Prof. Lijun Sun

Current works:

1. 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.
2. X. Chen, Z. Cheng, N. Saunier, L. Sun (2022). Laplacian convolutional representation for traffic time series imputation. arXiv preprint arXiv:2212.01529.

GitHub repositories:

1. **transdim**: Machine learning for spatiotemporal traffic data imputation and forecasting. (950 stars & 270 forks on GitHub) `https://github.com/xinychen/transdim`
2. **awesome-latex-drawing**: Academic drawing examples in LaTeX. (1,000 stars & 140 forks on GitHub) `https://github.com/xinychen/awesome-latex-drawing`

Slides:

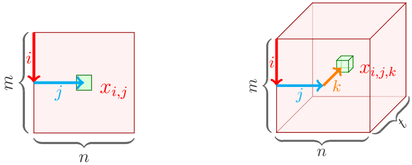
- `https://xinychen.github.io/slides/stdata_modeling.pdf`

Outline

- **Foundation: Tensor & Tensor Factorization**
  - What are Tensors?
  - Tensor Factorization (TF)
- **Time-Varying Autoregression**
  - Parameterize Coefficients via TF
  - Fluid Flow Data
  - Sea Surface Temperature Data
  - NYC Taxi Data
- **Low-Rank Laplacian Convolutional Model**
  - Reformulate Laplacian Regularization
  - Traffic Time Series Imputation
  - Multivariate Model for Speed Field Reconstruction
- **Matrix/Tensor Factorization**
  - Matrix Factorization
  - Smoothing Matrix Factorization
  - Hankel Tensor and Its Factorization
  - Spatiotemporal Hankel Tensor Factorization
  - Which Model Is Better?
- **Conclusion**

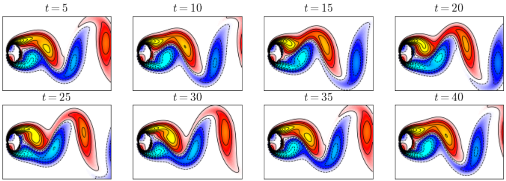
Foundation: Tensor & Tensor Factorization

- 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)

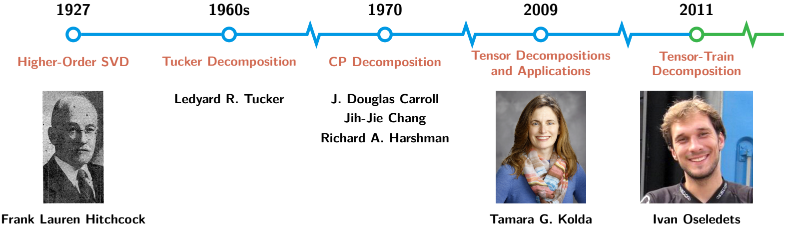
Foundation: Tensor & Tensor Factorization

- Revisit tensor factorization (TF)

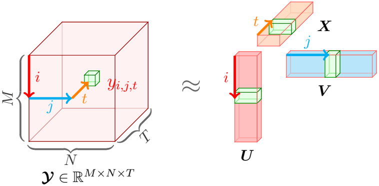
1927	1960s	1970	2009	2011
Higher-Order SVD	Tucker Decomposition	CP Decomposition	Tensor Decompositions and Applications	Tensor-Train Decomposition
 Frank Lauren Hitchcock	Ledyard R. Tucker	J. Douglas Carroll Jih-Jie Chang Richard A. Harshman	 Tamara G. Kolda	 Ivan Oseledets

Foundation: Tensor & Tensor Factorization

- Revisit tensor factorization (TF)



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



$$\begin{cases} y_{i,j,t} \approx \sum_{r=1}^{R} u_{i,r} v_{j,r} x_{t,r} \\ \boldsymbol{\mathcal{Y}} \approx \sum_{r=1}^{R} \boldsymbol{u}_r \otimes \boldsymbol{v}_r \otimes \boldsymbol{x}_r \end{cases}$$

Time-Varying Autoregression

- 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[1]:

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

- Alternating minimization

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

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

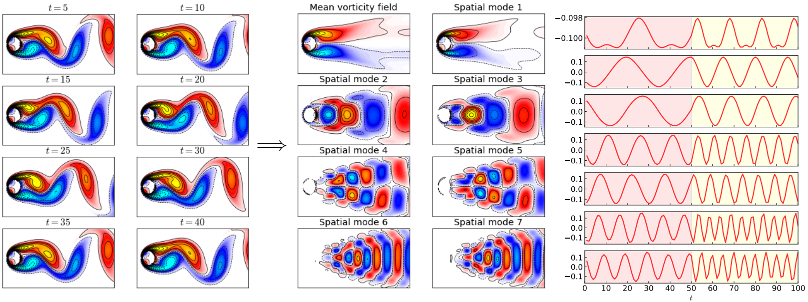
- Solve each subproblem by **conjugate gradient** or **least squares**.

[1] **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.

- Time-varying autoregression with TF

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

- **Fluid flow dataset** (the first 50 snapshots + 50 snapshots randomly selected from the last 100 snapshots)



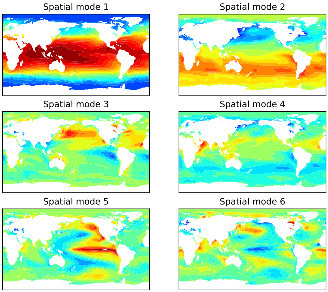
Vectorize snapshots as $\{\boldsymbol{y}_t\}$ Spatial modes in $\boldsymbol{W}$ Temporal modes in $\boldsymbol{X}$

- Produce interpretable patterns and identify the system of different frequencies.

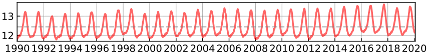
- Time-varying autoregression with TF

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

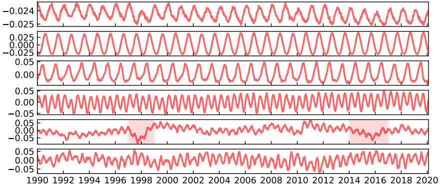
- Sea surface temperature (**SST**) dataset



Spatial modes in $\boldsymbol{W}$



Time series of mean temperature



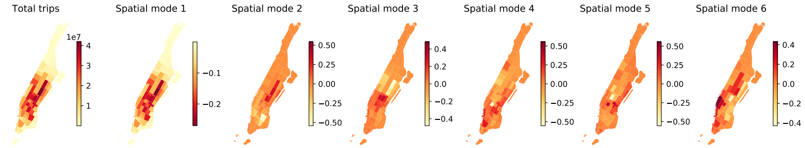
Temporal modes in $\boldsymbol{X}$

- Identify two strongest El Nino events (on 1997-98 & 2014-16).

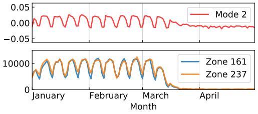
- Time-varying autoregression with TF

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

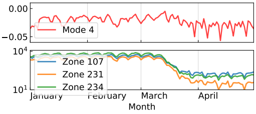
- **NYC taxi dataset** (pickup)



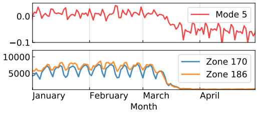
Spatial modes in $\boldsymbol{W}$



Mode #2 & taxi trips (2020)



Mode #4 & taxi trips (2020)

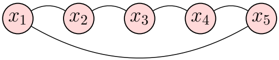


Mode #5 & taxi trips (2020)

Low-Rank Laplacian Convolutional Model

Reformulate Laplacian regularization with circular convolution.

- Intuition of (circulant) Laplacian matrix.



Undirected and circulant graph

$$\xrightarrow{\text{Modeling}} \quad \boldsymbol{L}=\begin{bmatrix} 2 & -1 & 0 & 0 & -1 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ -1 & 0 & 0 & -1 & 2 \end{bmatrix}$$

(Circulant) Laplacian matrix

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

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

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

- Local trend modeling via (Laplacian) temporal regularization:

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

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

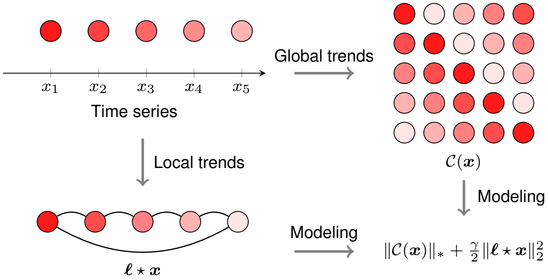
Low-Rank Laplacian Convolutional Model

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.



Low-Rank Laplacian Convolutional Model

- 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:

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

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

- The ADMM scheme:

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

Low-Rank Laplacian Convolutional Model

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

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

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

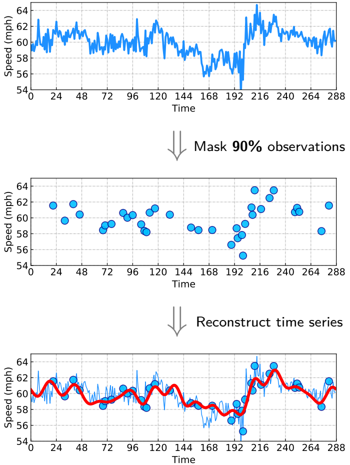
ℓ_1-norm Minimization in Complex Space (Liu & Zhang'22)

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

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

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

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



Speed (mph)
Time
Mask **90%** observations
Mask **95%** observations
Reconstruct time series
Reconstruct time series

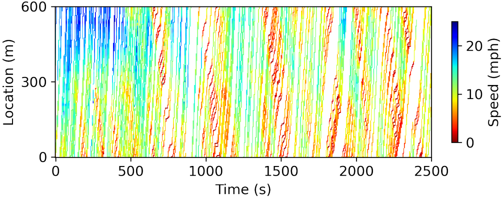
Low-Rank Laplacian Convolutional Model

Multivariate LCR (LCR-2D)

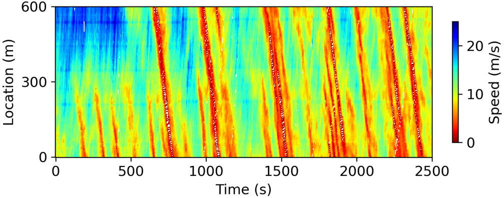
For any partially observed time series $\boldsymbol{Y}\in\mathbb{R}^{N\times T}$ with observed index set Ω, LCR can be formulated as follows,

$$\begin{aligned}\min_{\boldsymbol{X}}\ &\|\mathcal{C}(\boldsymbol{X})\|_{*}+\frac{\gamma}{2}\|(\boldsymbol{\ell}_s\boldsymbol{\ell}^\top)\star\boldsymbol{X}\|_F^2\\ \text{s.t. }&\|\mathcal{P}_{\Omega}(\boldsymbol{X}-\boldsymbol{Y})\|_F\leq\epsilon\end{aligned}$$

where $\mathcal{C}:\mathbb{R}^{N\times T}\to\mathbb{R}^{N\times N\times T\times T}$ denotes the circulant operator.

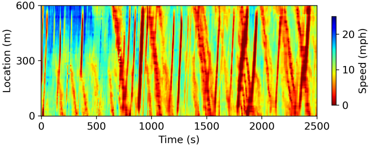


200-by-500 matrix (NGSIM) $\Downarrow$ Reconstruct speed field from 20% sparse trajectories?



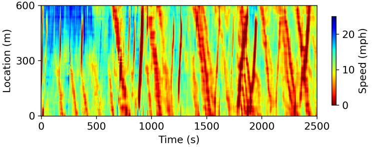
- How to learn from sparse spatiotemporal data?
- How to characterize spatial/temporal local dependencies?

Quadratic variation completion (QVC)



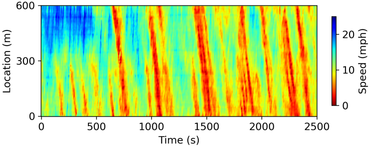
MAPE = 51.50% & RMSE = 4.86mph

Laplacian kernelized completion (LKC)



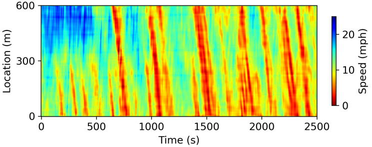
MAPE = 46.94% & RMSE = 4.34mph

Circulant tensor nuclear norm minimization (CTNNM)



MAPE = 43.51% & RMSE = 1.65mph

Laplacian convolutional representation (LCR)



MAPE = **41.29%** & RMSE = **1.55mph**

- QVC & LKC:

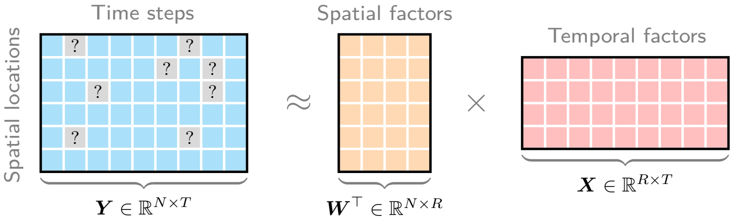
$$\begin{aligned}\min_{\boldsymbol{X}}\ & \frac{\gamma}{2}\|(\boldsymbol{\ell}_s\boldsymbol{\ell}^\top)\star\boldsymbol{X}\|_F \\ \text{s.t. } & \|\mathcal{P}_\Omega(\boldsymbol{X}-\boldsymbol{Y})\|_F\leq\epsilon\end{aligned}$$

- CTNNM:

$$\begin{aligned}\min_{\boldsymbol{X}}\ & \|\mathcal{C}(\boldsymbol{X})\|_* \\ \text{s.t. } & \|\mathcal{P}_\Omega(\boldsymbol{X}-\boldsymbol{Y})\|_F\leq\epsilon\end{aligned}$$

Matrix/Tensor Factorization

- 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 $\boldsymbol{X}$. ($\|\cdot\|_F^2$ is the squared Frobenius norm.)

  - Objective function $f(\boldsymbol{W},\boldsymbol{X})$ or f;
  - Rank $R \in \mathbb{N}^{+}$ $(R < \min\{N, T\})$;
  - Orthogonal projection $\mathcal{P}_{\Omega}(\cdot)$.

Matrix/Tensor Factorization

- MF optimization problem

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

- Partial derivatives

$$\begin{cases}\dfrac{\partial f}{\partial \boldsymbol{W}}=-\boldsymbol{X}\mathcal{P}_{\Omega}^{\top}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})+\rho\boldsymbol{W}\\ \dfrac{\partial f}{\partial \boldsymbol{X}}=-\boldsymbol{W}\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})+\rho\boldsymbol{X}\end{cases}$$

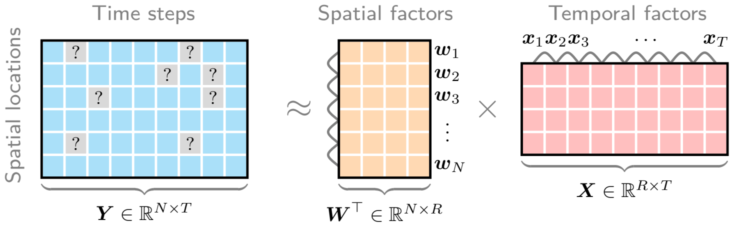
- Alternating least squares (**ALS**)

$$\begin{cases}\dfrac{\partial f}{\partial \boldsymbol{W}}=\mathbf{0}\\ \dfrac{\partial f}{\partial \boldsymbol{X}}=\mathbf{0}\end{cases}\Longrightarrow\begin{cases}\boldsymbol{w}_i:=\Big(\displaystyle\sum_{t:(i,t)\in\Omega}\boldsymbol{x}_t\boldsymbol{x}_t^{\top}+\rho\boldsymbol{I}_R\Big)^{-1}\sum_{t:(i,t)\in\Omega}\boldsymbol{x}_t y_{i,t}\\ \boldsymbol{x}_t:=\Big(\displaystyle\sum_{i:(i,t)\in\Omega}\boldsymbol{w}_i\boldsymbol{w}_i^{\top}+\rho\boldsymbol{I}_R\Big)^{-1}\sum_{i:(i,t)\in\Omega}\boldsymbol{w}_i y_{i,t}\end{cases}$$

- Latent factors
  - $\boldsymbol{w}_i\in\mathbb{R}^R$, $i=1,2,\ldots,N$ are the columns of $\boldsymbol{W}$;
  - $\boldsymbol{x}_t\in\mathbb{R}^R$, $t=1,2,\ldots,T$ are the columns of $\boldsymbol{X}$.

Smoothing matrix factorization

- Spatial/temporal local dependencies are also important!



- Formulate spatial/temporal dependencies

$$\boldsymbol{W}\boldsymbol{\Psi}_1^\top = \begin{bmatrix} | & & | \\ \boldsymbol{w}_2 - \boldsymbol{w}_1 & \cdots & \boldsymbol{w}_N - \boldsymbol{w}_{N-1} \\ | & & | \end{bmatrix}$$

$$\boldsymbol{X}\boldsymbol{\Psi}_2^\top = \begin{bmatrix} | & & | \\ \boldsymbol{x}_2 - \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_T - \boldsymbol{x}_{T-1} \\ | & & | \end{bmatrix}$$

Matrix/Tensor Factorization

Smoothing Matrix Factorization

- Formulate spatial/temporal dependencies

$$\boldsymbol{\Psi}=\begin{bmatrix}-1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1\end{bmatrix} \Longrightarrow \begin{cases}\|\boldsymbol{W}\boldsymbol{\Psi}_1^\top\|_F^2 & \text{with } \boldsymbol{\Psi}_1\in\mathbb{R}^{(N-1)\times N} \\ \|\boldsymbol{X}\boldsymbol{\Psi}_2^\top\|_F^2 & \text{with } \boldsymbol{\Psi}_2\in\mathbb{R}^{(T-1)\times T}\end{cases}$$

- SMF optimization problem

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{X}}\ &\frac{1}{2}\left\|\mathcal{P}_\Omega(\boldsymbol{Y}-\boldsymbol{W}^\top\boldsymbol{X})\right\|_F^2+\frac{\rho}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2) \\ &+\frac{\lambda}{2}(\|\boldsymbol{W}\boldsymbol{\Psi}_1^\top\|_F^2+\|\boldsymbol{X}\boldsymbol{\Psi}_2^\top\|_F^2)\end{aligned}$$

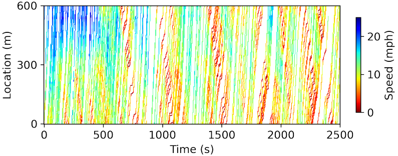
- **Alternating minimization**

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

- Solve each matrix equation by the **conjugate gradient** method.

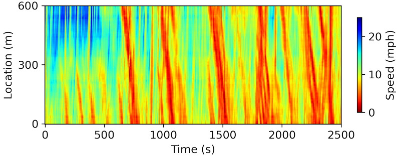
Matrix/Tensor Factorization

- Speed field reconstruction
  - Set rank $R = 10$, weight parameter $\rho = 10$.



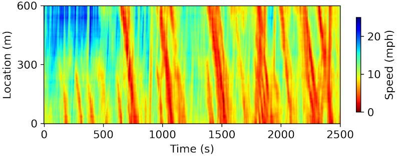
Sparse speed field

MF



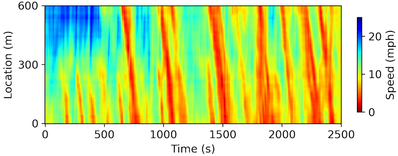
MAPE = 45.84%, RMSE = 2.80mph

SMF ($\lambda = 10$)



MAPE = **44.06%**, RMSE = 2.16mph

SMF ($\lambda = 10^2$)



MAPE = 48.00%, RMSE = **1.60mph**

Hankel tensor and its 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 $\tau = 2$:

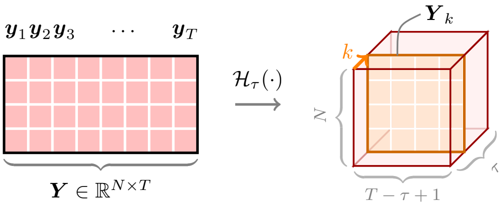
$$\mathcal{H}_\tau(\boldsymbol{y}) = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \\ y_3 & y_4 \\ y_4 & y_5 \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

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

  - Automatic temporal modeling.

Hankel tensor and its factorization

- (Hankelization) 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: $N \times (T-\tau+1)$.



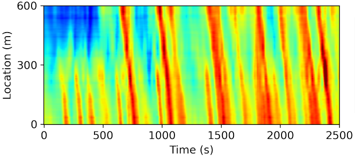
Hankel tensor and its factorization

- HTF optimization problem

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{X}} \ \frac{1}{2}\left\|\mathcal{P}_{\tilde{\Omega}}\left(\mathcal{H}_{\tau}(\boldsymbol{Y})-\sum_{r=1}^{R}\boldsymbol{u}_r\otimes\boldsymbol{v}_r\otimes\boldsymbol{x}_r\right)\right\|_F^2$$

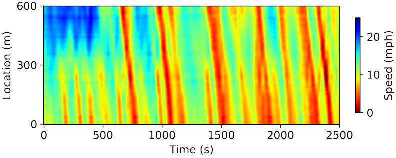
- HTF's advantage/disadvantage over MF:
  - ✓ Automatic temporal modeling
  - ✗ High memory consumption
- Speed field reconstruction
  - Set rank $R=10$;
  - Recall that SMF: MAPE = 48.00% & RMSE = 1.60mph.

HTF ($\tau=10$)



MAPE = **41.40%**, RMSE = **1.42mph**

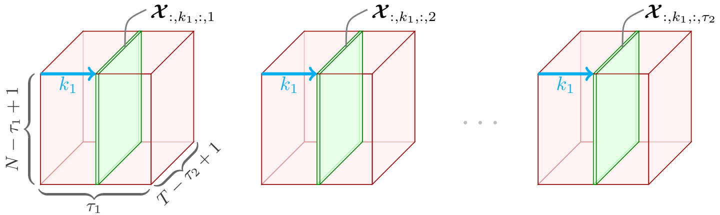
HTF ($\tau=15$)



MAPE = 43.97%, RMSE = **1.42mph**

Spatiotemporal Hankel tensor factorization

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



- StHTF optimization problem

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

Matrix/Tensor Factorization

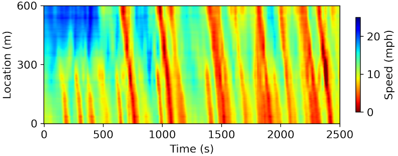
Spatiotemporal Hankel tensor factorization

- StHTF optimization problem

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

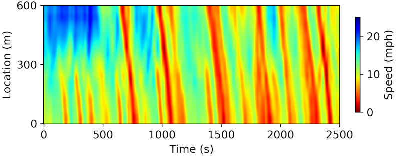
- Speed field reconstruction
    - Set rank $R=10$;
    - Recall that SMF: MAPE $=$ 48.00% & RMSE $=$ 1.60mph.

HTF ($\tau=10$)



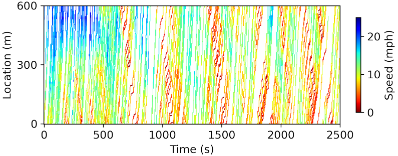
MAPE $=$ **41.40%**, RMSE $=$ 1.42mph

StHTF ($\tau_1=\tau_2=10$)

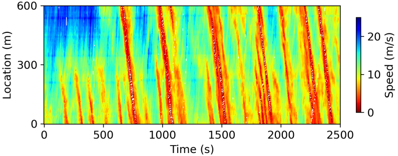


MAPE $=$ 41.58%, RMSE $=$ **1.39mph**

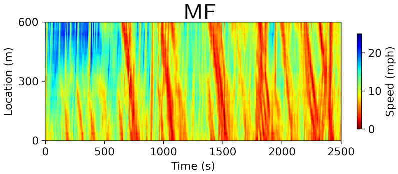
Which Model Is Better?

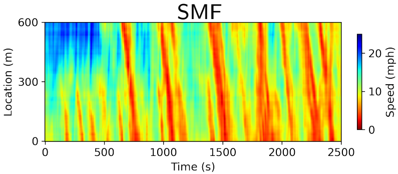


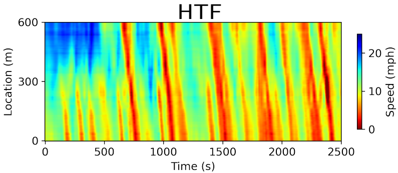
Sparse speed field



Ground truth speed field

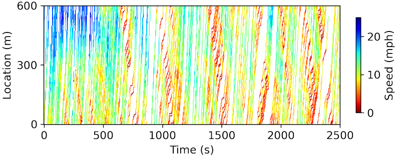




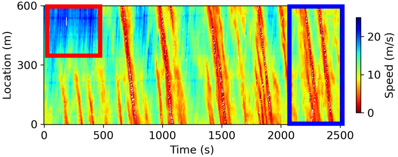


Matrix/Tensor Factorization

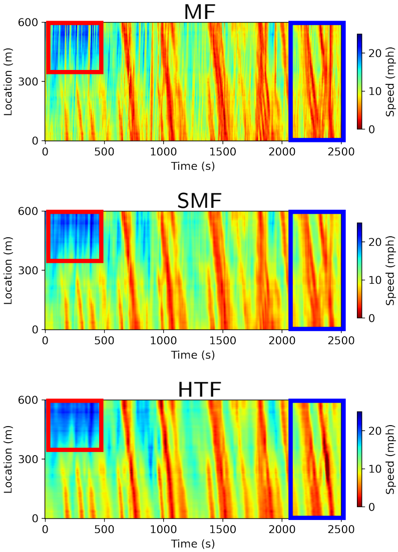
Which Model Is Better?



Sparse speed field



Ground truth speed field



Conclusion

- Time-varying autoregression
  - use TF to compress the coefficients
  - interpret factorization structure as spatial/temporal modes
- Low-rank Laplacian convolutional representation
  - characterize local trends of time series via the Laplacian kernels
  - solve the model by a fast FFT implementation
- Spatiotemporal matrix/tensor factorization
  - Smoothing regularization
  - Hankelization

Thanks for your attention!

Any Questions?

About me:

- Homepage: https://xinychen.github.io
- Google Scholar: user=mCrW04wAAAAJhl (**650+ citations**)
- GitHub: https://github.com/xinychen (**3.2k+ stars**)
- Blog: https://medium.com/@xinyu.chen (**65k+ views**)
- How to reach me: chenxy346@gmail.com