



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

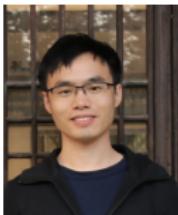


Low-Rank Matrix and Tensor Methods for Multidimensional Data Modeling

Xinyu Chen

University of Montreal, Canada

July 28, 2023



Ph.D. Candidate
Xinyu Chen



Supervisor
Prof. Nicolas Saunier



Co-supervisor
Prof. Lijun Sun (McGill)

Recent studies:

- ❶ X. Chen, C. Zhang, X. Chen, N. Saunier, L. Sun (2023). Discovering dynamic patterns from spatiotemporal data with time-varying low-rank autoregression. IEEE Transactions on Knowledge and Data Engineering. Early access.
- ❷ X. Chen, Z. Cheng, N. Saunier, L. Sun (2022). Laplacian convolutional representation for traffic time series imputation. arXiv preprint arXiv:2212.01529.
- 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.

GitHub repository:

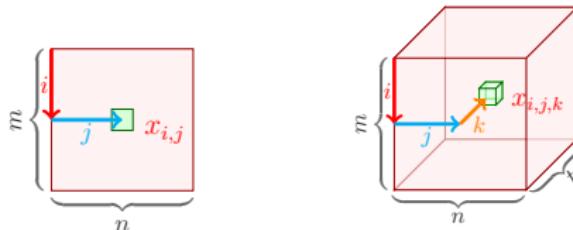
- **transdim**: Machine learning for spatiotemporal traffic data imputation and forecasting. (1,000 stars & 270 forks on GitHub)
<https://github.com/xinychen/transdim>

Slides:

- https://xinychen.github.io/slides/tensors_in10min.pdf

Tensors

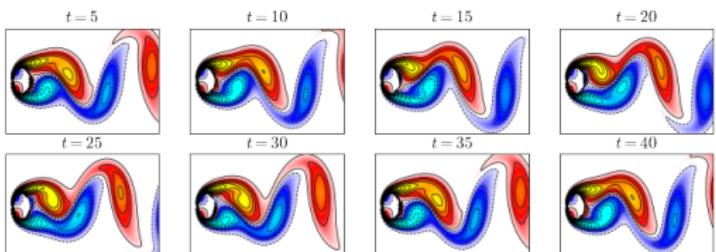
- What is tensor? $\mathbf{X} \in \mathbb{R}^{m \times n}$ vs. $\mathcal{X} \in \mathbb{R}^{m \times n \times t}$



- Tensors are everywhere!



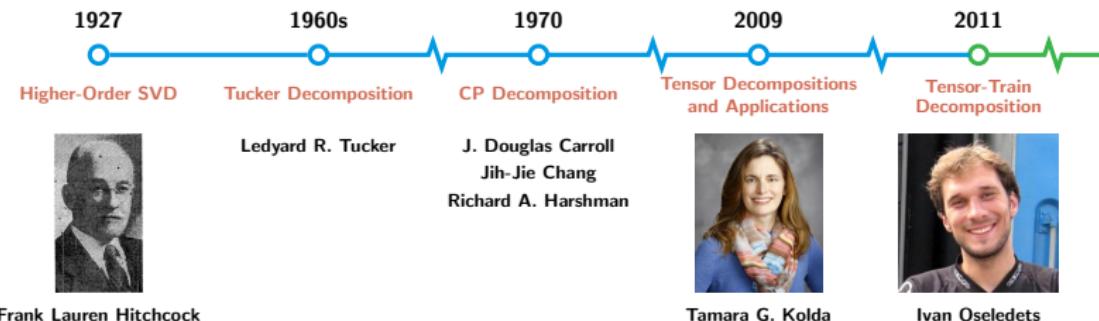
Color image with
RGB channels



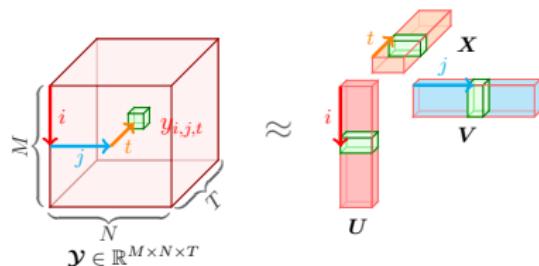
Dynamical system (fluid flow)

Tensor Factorization

- Revisit tensor factorization (TF)



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



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

Time-Varying Autoregression

- Given a sequence of time series $\mathbf{y}_t \in \mathbb{R}^N$, $t = 1, 2, \dots, T$,

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

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

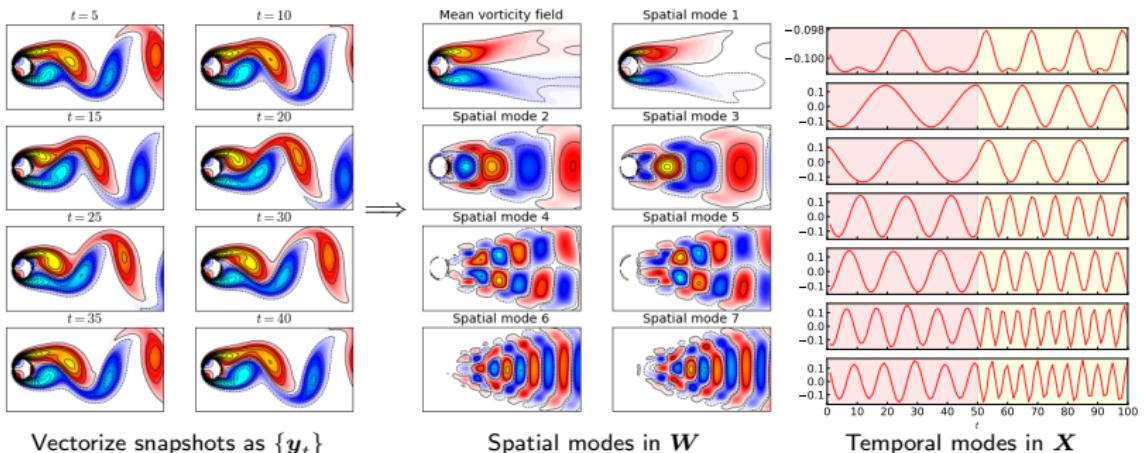
- (Ours)** Parameterize coefficients via TF:

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

- Time-varying autoregression with TF

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

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

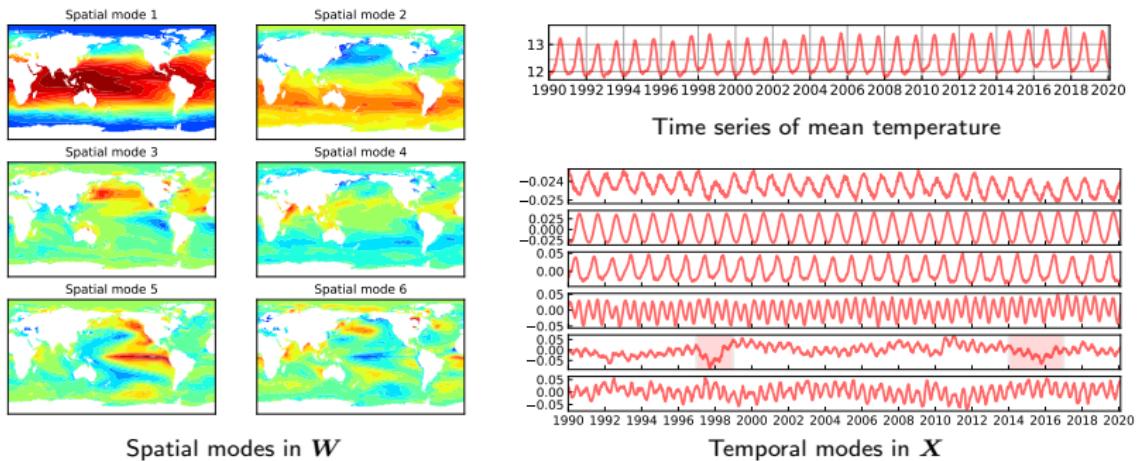


- Produce interpretable patterns and identify the system of different frequencies.

- Time-varying autoregression with TF

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

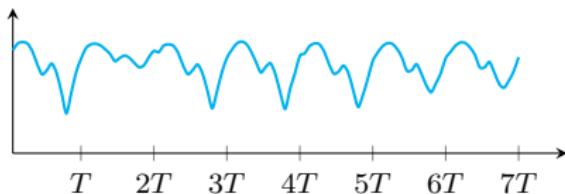
- Sea surface temperature (**SST**) dataset



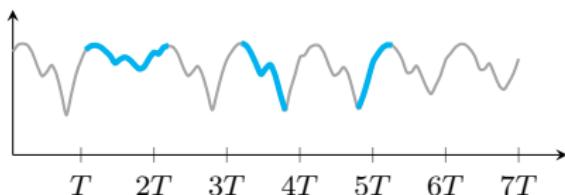
- Identify two strongest El Nino events (on 1997-98 & 2014-16).

Motivation:

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



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



- [Question] How to characterize both global and local trends in sparse time series data?

Laplacian Convolutional Representation

Reformulate Laplacian regularization with circular convolution.

- Intuition of (circulant) Laplacian matrix.

The diagram illustrates the mapping between a graph and its corresponding matrix. On the left, an "Undirected and circulant graph" is shown with five nodes labeled x_1 , x_2 , x_3 , x_4 , and x_5 . Edges connect adjacent nodes (x_1 to x_2 , x_2 to x_3 , x_3 to x_4 , x_4 to x_5) and a self-loop on node x_2 . An arrow labeled "Modeling" points to the right, where the "(Circulant) Laplacian matrix" is given as:

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

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

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

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

- Local trend modeling via (Laplacian) temporal regularization:

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

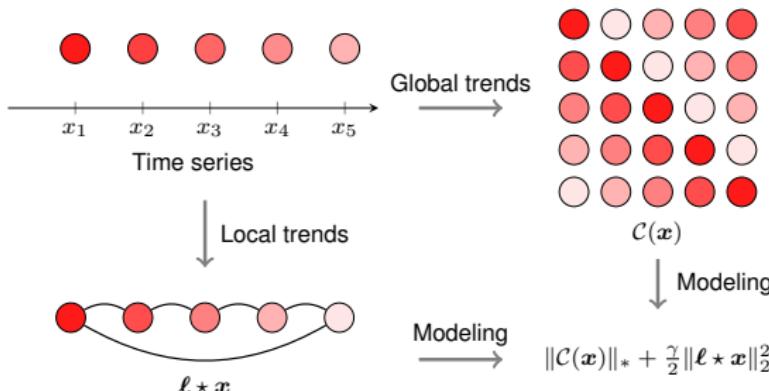
Laplacian Convolutional Representation

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.,

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

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



Laplacian Convolutional Representation

- LCR model:

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

- Augmented Lagrangian function:

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

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

- The ADMM scheme:

$$\left\{ \begin{array}{l} \mathbf{x} := \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) \\ \mathbf{z} := \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}, \mathbf{z}, \mathbf{w}) \\ = \frac{1}{\lambda + \eta} \mathcal{P}_\Omega(\lambda \mathbf{x} + \mathbf{w} + \eta \mathbf{y}) + \frac{1}{\lambda} \mathcal{P}_\Omega^\perp(\lambda \mathbf{x} + \mathbf{w}) \\ \mathbf{w} := \mathbf{w} + \lambda(\mathbf{x} - \mathbf{z}) \end{array} \right.$$

Laplacian Convolutional Representation

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

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

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).

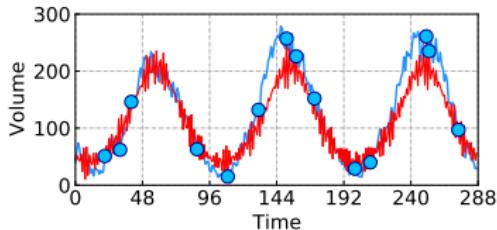
ℓ_1 -norm Minimization in Complex Space (Liu & Zhang'23)

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

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

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

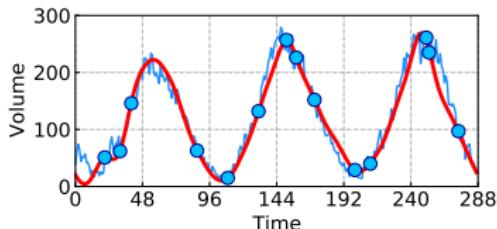
$$\hat{x}_t := \frac{\hat{h}_t}{|\hat{h}_t|} \cdot \max\{0, |\hat{h}_t| - T/\lambda\}, t = 1, \dots, T.$$



CircNNM:

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

↓ Plus temporal regularization (TR)

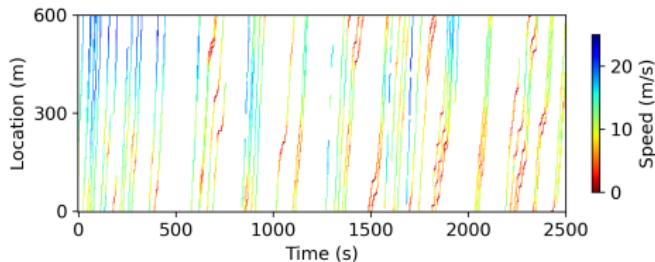


LCR:

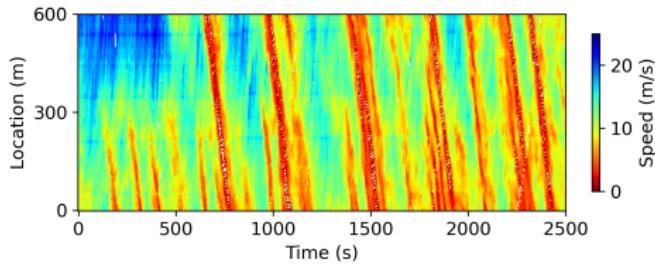
$$\begin{aligned} & \min_{\boldsymbol{x}} \|\mathcal{C}(\boldsymbol{x})\|_* + \frac{\gamma}{2} \|\boldsymbol{\ell} * \boldsymbol{x}\|_2^2 \\ \text{s. t. } & \|\mathcal{P}_\Omega(\boldsymbol{x} - \boldsymbol{y})\|_2 \leq \epsilon \end{aligned}$$

Motivation:

- Speed field reconstruction problem in vehicular traffic flow.



200-by-500 matrix
(NGSIM) ↓ Reconstruct speed field from
 5% sparse trajectories?



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

Hankel Tensor Factorization

- Hankel matrix

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

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

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

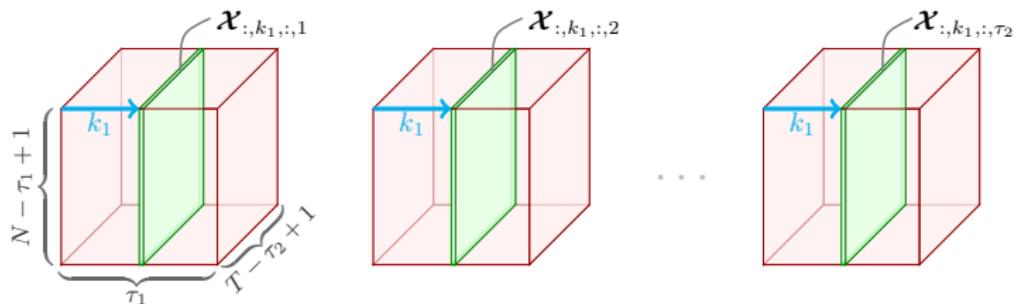
$$\mathcal{H}_\tau(\mathbf{y}) = \begin{bmatrix} y_1 & y_2 \\ y_2 & y_3 \\ y_3 & y_4 \\ y_4 & y_5 \end{bmatrix} \approx \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \otimes \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\Rightarrow \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \hat{y}_3 \\ \hat{y}_4 \\ \hat{y}_5 \end{bmatrix} = \mathcal{H}_\tau^{-1} \left(\begin{bmatrix} v_1 x_1 & v_1 x_2 \\ v_2 x_1 & v_2 x_2 \\ v_3 x_1 & v_3 x_2 \\ v_4 x_1 & v_4 x_2 \end{bmatrix} \right) = \begin{bmatrix} v_1 x_1 \\ (v_1 x_2 + v_2 x_1)/2 \\ (v_2 x_2 + v_3 x_1)/2 \\ (v_3 x_2 + v_4 x_1)/2 \\ v_4 x_2 \end{bmatrix}$$

- Automatic temporal modeling.

Hankel Tensor Factorization

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



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

Hankel Tensor Factorization

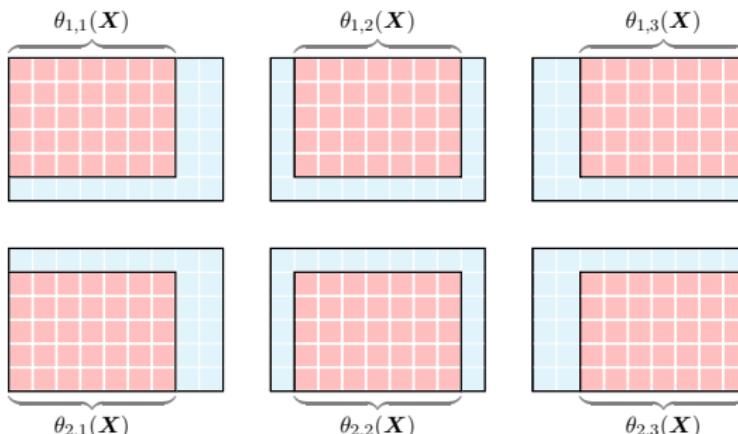
Hankel indexing:

- Sampling function for the Hankelization:

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

referring to 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

Our model:

- 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:

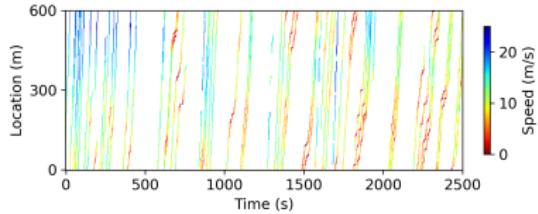
- CP tensor decomposition (Khatri-Rao product \odot):

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

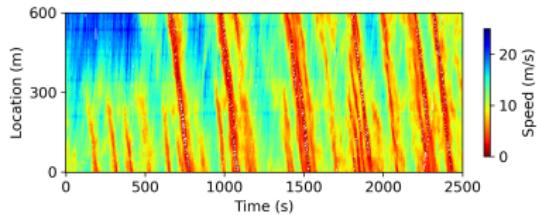
- Tensor-train decomposition¹:

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

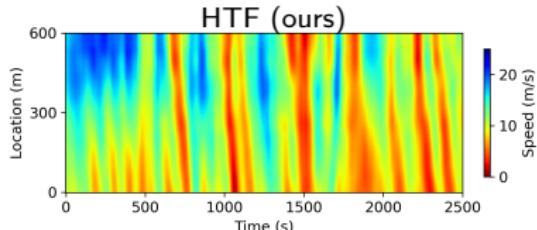
¹If $\{\mathbf{S}_{k_1}, \mathbf{V}_{k_2}\}$ are circulant matrices, then tensor-train produces our model; if $\{\mathbf{S}_{k_1}, \mathbf{V}_{k_2}\}$ are diagonal matrices, then tensor-train produces CP model.



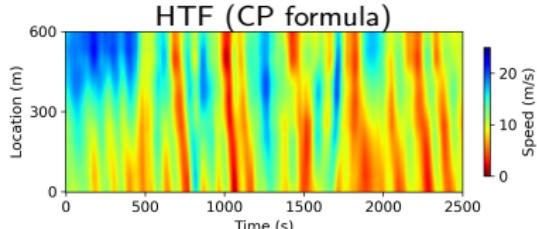
Sparse speed field



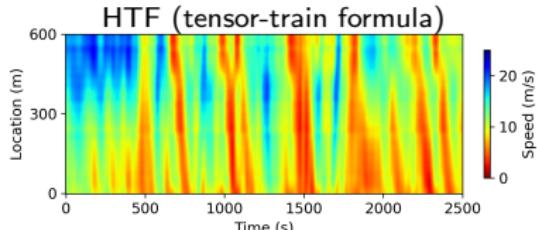
Ground truth speed field



MAPE = 51.92%



MAPE = 53.93%



MAPE = 56.48%



POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉNIERIE



Thanks for your attention!

Any Questions?

About me:

- 🏠 Homepage: <https://xinychen.github.io>
- 👤 GitHub: <https://github.com/xinychen> (3.4k+ stars)
- 💻 Blog: <https://medium.com/@xinyu.chen> (70k+ views)
- ✉️ How to reach me: chenxy346@gmail.com