



POLYTECHNIQUE
MONTRÉAL

UNIVERSITÉ
D'INGÉNIERIE

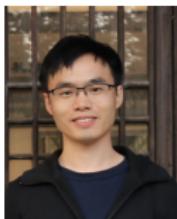


Nonstationary Temporal Matrix Factorization for Multivariate Time Series Forecasting

Xinyu Chen (Ph.D. candidate)

Polytechnique Montréal & Université de Montréal

May 11, 2022



Xinyu Chen
PolyMtl



Chengyuan Zhang
McGill



Xi-Le Zhao
UESTC



Nicolas Saunier
PolyMtl



Lijun Sun
McGill

Source

- ❶ **Preprint:** Xinyu Chen, Chengyuan Zhang, Xi-Le Zhao, Nicolas Saunier, Lijun Sun (2022). Nonstationary temporal matrix factorization for multivariate time series forecasting. arXiv preprint arXiv:2203.10651.
- ❷ **Data & Python code:** <https://github.com/xinyuchen/tracebase>
- ❸ **Blog post:** <https://medium.com/@xinyu.chen>
- ❹ **Slides:** <https://xinyuchen.github.io/slides/notmf.pdf>

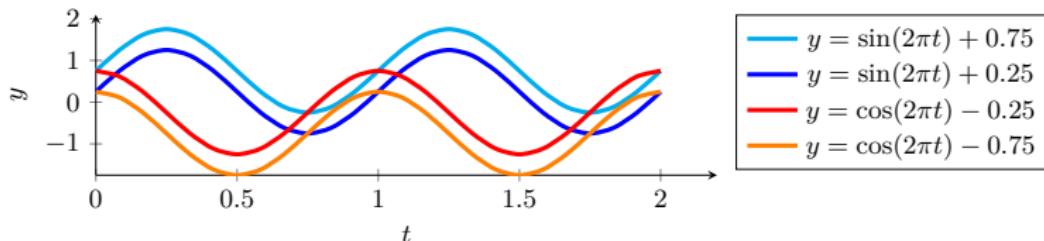
Outline

- **Motivation** (high-dimensionality, sparsity, nonstationarity)
- **Objective** (time series forecasting)
- **Methodology** ($\text{MF} \rightarrow \text{TMF} \rightarrow \text{NoTMF}$)
- **Uber Movement Data** (NYC & Seattle)
- **Experiments**
- **Concluding Remarks**

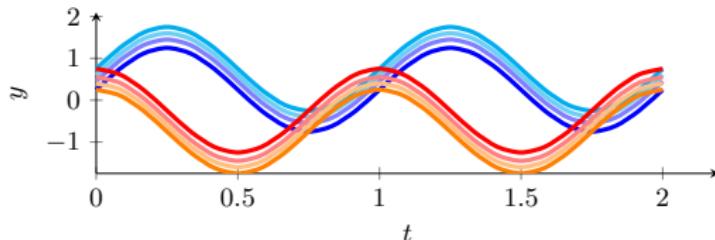
Motivation

[M1] High-dimensional time series could stem from a relatively small number of latent temporal factors.

- Four sequences with two kinds of temporal dynamics (Sine & Cosine).



- Several sequences with two kinds of temporal dynamics.

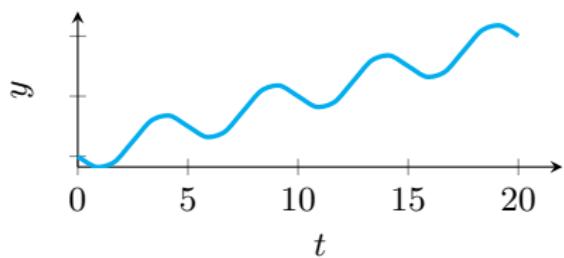


Methodology

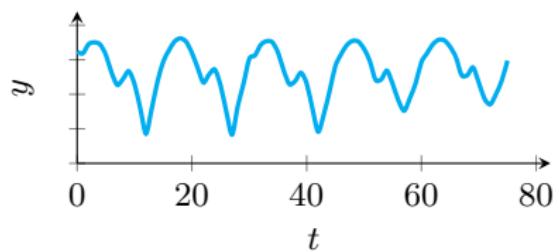
[M2] Sparse time series may reveal low-rank patterns.

[M3] Modern time series usually show nonstationarity (e.g., trend, seasonality).

Trends & changing levels



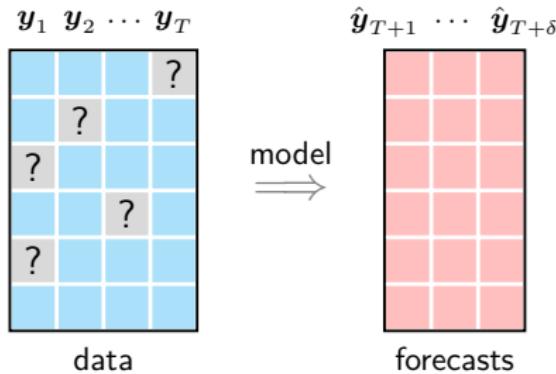
Seasonality



Objective

Multivariate time series forecasting

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



[Q]

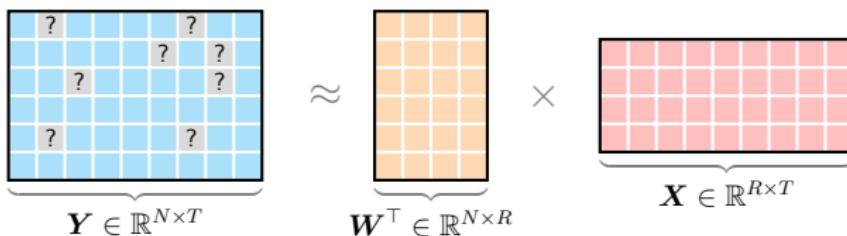
- How to learn from *high-dimensional* and *sparse* data?
- How to model *nonstationarity* in time series?
- How to perform forecasting on these time series?

Methodology

Low-rank matrix factorization (Koren et al.'09)

Given any partially observed data matrix $\mathbf{Y} \in \mathbb{R}^{N \times T}$ with observed index set Ω , then the optimization problem of matrix factorization with rank R can be formulated as

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



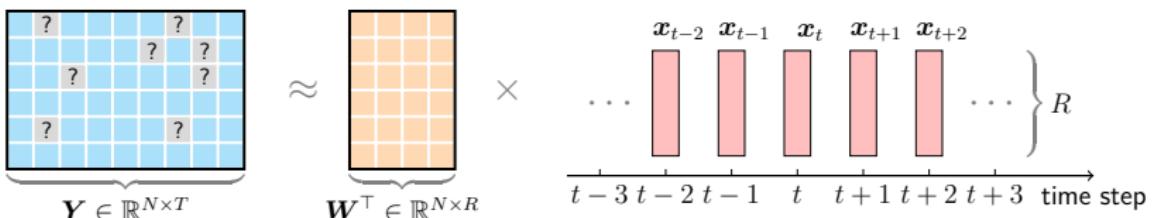
- (😢) Cannot capture temporal correlations.
- (😢) Cannot perform time series forecasting.

Methodology

Temporal matrix factorization (Yu et al.'16; Chen & Sun'21)

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

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



VAR is usually built on stationary time series (temporal factors).

Methodology

Nonstationary temporal matrix factorization (NoTMF)

Given any partially observed time series data $\mathbf{Y} \in \mathbb{R}^{N \times T}$ with observed index set Ω , then we assume a season- m differencing on the latent temporal factors:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{X}, \{\mathbf{A}_k\}_{k=1}^d} & \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) \\ & + \frac{\lambda}{2} \sum_{t=d+m+1}^T \|(\mathbf{x}_t - \mathbf{x}_{t-m}) - \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k})\|_2^2 \end{aligned} \quad (3)$$

- First-order differencing $\mathbf{x}'_t = \mathbf{x}_t - \mathbf{x}_{t-1}$.
 - Second-order differencing $\mathbf{x}''_t = (\mathbf{x}_t - \mathbf{x}_{t-1}) - (\mathbf{x}_{t-1} - \mathbf{x}_{t-2})$.
 - Twice-differenced series $\mathbf{x}'''_t = (\mathbf{x}_t - \mathbf{x}_{t-m}) - (\mathbf{x}_{t-1} - \mathbf{x}_{t-m-1})$.
- 😊 Stationarizing a time series with differencing can improve the prediction.¹

¹Stationarity and differencing: <https://otexts.com/fpp2/stationarity.html>

Methodology

Rewrite VAR in the form of matrix

Temporal operators

For any multivariate time series $\mathbf{X} \in \mathbb{R}^{R \times T}$ with $m, d \in \mathbb{N}^+$, if we define temporal operators as

$$\begin{aligned}\Psi_k &\triangleq \begin{bmatrix} \mathbf{0}_{(T-d-m) \times (d-k)} & -\mathbf{I}_{T-d-m} & \mathbf{0}_{(T-d-m) \times (k+m)} \\ + \begin{bmatrix} \mathbf{0}_{(T-d-m) \times (d+m-k)} & \mathbf{I}_{T-d-m} & \mathbf{0}_{(T-d-m) \times k} \end{bmatrix} \\ \in \mathbb{R}^{(T-d-m) \times T}, k = 0, 1, \dots, d \end{aligned} \quad (4)$$

then

$$\begin{aligned}&\sum_{t=d+m+1}^T \|(\mathbf{x}_t - \mathbf{x}_{t-m}) - \sum_{k=1}^d \mathbf{A}_k (\mathbf{x}_{t-k} - \mathbf{x}_{t-m-k})\|_2^2 \\ &\equiv \|\mathbf{X} \Psi_0^\top - \sum_{k=1}^d \mathbf{A}_k \mathbf{X} \Psi_k^\top\|_F^2 \triangleq \|\mathbf{X} \Psi_0^\top - \mathbf{A} (\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top\|_F^2 \quad (5)\end{aligned}$$

where $\mathbf{A} \triangleq [\mathbf{A}_1 \quad \cdots \quad \mathbf{A}_d]$ and $\Psi \triangleq [\Psi_1 \quad \cdots \quad \Psi_d]$.

Methodology

Rewrite NoTMF:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{X}, \mathbf{A}} \quad & \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})\|_F^2 + \frac{\rho}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{X}\|_F^2) \\ & + \frac{\lambda}{2} \|\mathbf{X} \Psi_0^\top - \mathbf{A} (\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top\|_F^2 \end{aligned} \tag{6}$$

Alternating minimization method:

- w.r.t. \mathbf{W} :

$$\frac{\partial f}{\partial \mathbf{W}} = -\mathbf{X} \mathcal{P}_\Omega^\top (\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{W} = \mathbf{0}$$

- w.r.t. \mathbf{X} :

$$\frac{\partial f}{\partial \mathbf{X}} = -\mathbf{W} \mathcal{P}_\Omega (\mathbf{Y} - \mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} + \lambda \sum_{k=0}^d \mathbf{A}_k^\top \left(\sum_{h=0}^d \mathbf{A}_h \mathbf{X} \Psi_h^\top \right) \Psi_k = \mathbf{0}$$

- w.r.t. \mathbf{A} :

$$\mathbf{A} = \mathbf{X} \Psi_0^\top [(\mathbf{I}_d \otimes \mathbf{X}) \Psi^\top]^\dagger$$

Methodology

Solve generalized Sylvester equation (w.r.t. \mathbf{X}):

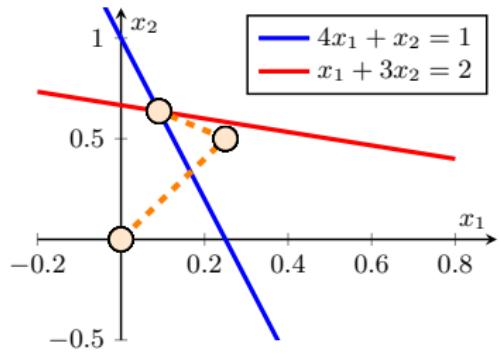
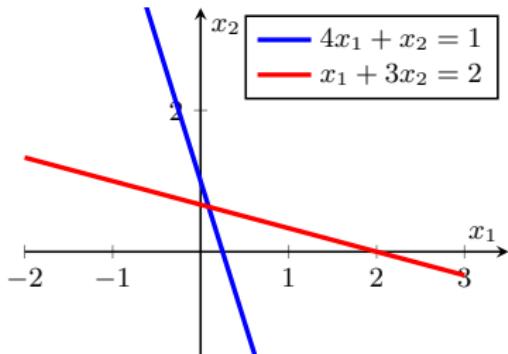
$$\underbrace{\mathbf{W}\mathcal{P}_\Omega(\mathbf{W}^\top \mathbf{X}) + \rho \mathbf{X} + \lambda \sum_{k=0}^d \mathbf{A}_k^\top \left(\sum_{h=0}^d \mathbf{A}_h \mathbf{X} \boldsymbol{\Psi}_h^\top \right) \boldsymbol{\Psi}_k}_{\mathcal{L}_x(\mathbf{X}) \triangleq \text{vec}(\cdot)} = \mathbf{W}\mathcal{P}_\Omega(\mathbf{Y})$$

- Conjugate gradient for inferring \mathbf{X} :

- Initialize \mathbf{x} as \mathbf{x}_0 , and compute the residual
 $r_0 := \text{vec}(\mathbf{W}\mathcal{P}_\Omega(\mathbf{Y})) - \mathcal{L}_x(\mathbf{X}_0)$. Let $\mathbf{q}_0 := \mathbf{r}_0$.
- Repeat:
 - Convert \mathbf{q}_ℓ into matrix \mathbf{Q}_ℓ .
 - Compute $\alpha_\ell := \frac{\mathbf{r}_\ell^\top \mathbf{r}_\ell}{\mathbf{q}_\ell^\top \mathcal{L}_x(\mathbf{Q}_\ell)}$, and update
$$\begin{cases} \mathbf{x}_{\ell+1} := \mathbf{x}_\ell + \alpha_\ell \mathbf{q}_\ell \\ \mathbf{r}_{\ell+1} := \mathbf{r}_\ell - \alpha_\ell \mathcal{L}_x(\mathbf{Q}_\ell) \end{cases}.$$
 - Compute $\beta_\ell := \frac{\mathbf{r}_{\ell+1}^\top \mathbf{r}_{\ell+1}}{\mathbf{r}_\ell^\top \mathbf{r}_\ell}$, and update $\mathbf{q}_{\ell+1} := \mathbf{r}_{\ell+1} + \beta_\ell \mathbf{q}_\ell$.
 - Update $\ell := \ell + 1$.

Methodology

- [Q] $Ax = b$ with known $A \in \mathbb{R}^n, b \in \mathbb{R}^n$, the solution of $x \in \mathbb{R}^n$?
- **Conjugate gradient** method:
 - Initialize x as x_0 , and compute the residual $r_0 := b - Ax_0$. Let $q_0 := r_0$.
 - Repeat:
 - Compute $\alpha_\ell := \frac{r_\ell^\top r_\ell}{q_\ell^\top A q_\ell}$, and update $\begin{cases} x_{\ell+1} := x_\ell + \alpha_\ell q_\ell \\ r_{\ell+1} := r_\ell - \alpha_\ell A q_\ell \end{cases}$.
 - Compute $\beta_\ell := \frac{r_{\ell+1}^\top r_{\ell+1}}{r_\ell^\top r_\ell}$, and update $q_{\ell+1} := r_{\ell+1} + \beta_\ell q_\ell$.
 - Update $\ell := \ell + 1$.



Methodology

NoTMF forecasting on streaming data?

- NoTMF: Use \mathbf{Y}_t to estimate $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$.

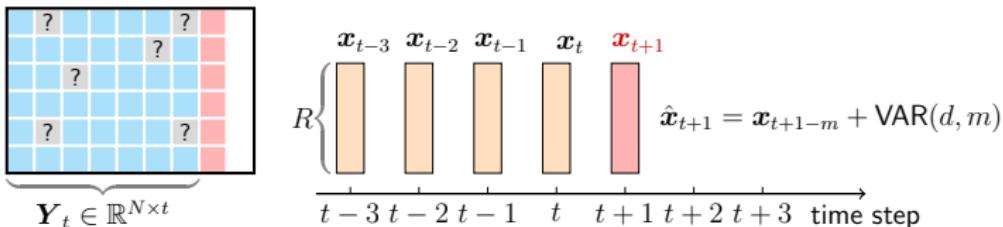
$$\underbrace{\begin{matrix} ? & & & ? & \\ & ? & & & \\ ? & & ? & & \\ & & & & \end{matrix}}_{\mathbf{Y}_t \in \mathbb{R}^{N \times t}}$$

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

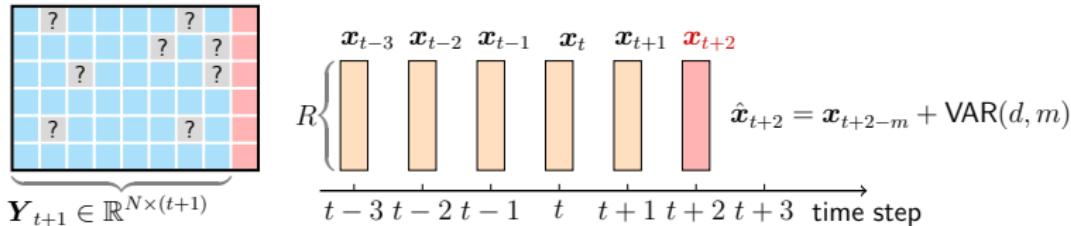
Methodology

NoTMF forecasting on streaming data?

- NoTMF: Use \mathbf{Y}_t to estimate $\{\mathbf{W}, \mathbf{X}, \mathbf{A}\}$.



- Online forecasting (Gultekin & Paisley'18): Fix \mathbf{W} and use \mathbf{Y}_{t+1} to update $\{\mathbf{X}, \mathbf{A}\}$.



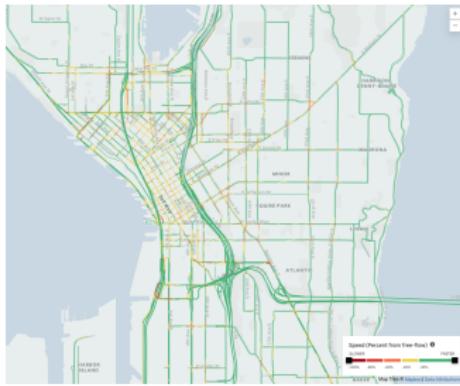
Uber Movement Data

High-dimensionality & Sparsity

- Uber (hourly) movement speed data²



NYC movement



Seattle movement

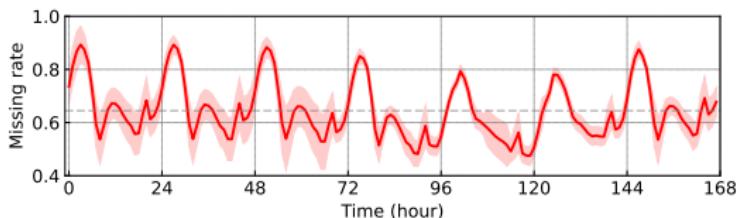
- The average speed on a given road segment for each hour of each day.
- Hourly speeds are computed when road segments have 5+ unique trips.
- **Issue:** insufficient sampling of ridesharing vehicles on the road network.

²<https://movement.uber.com/>

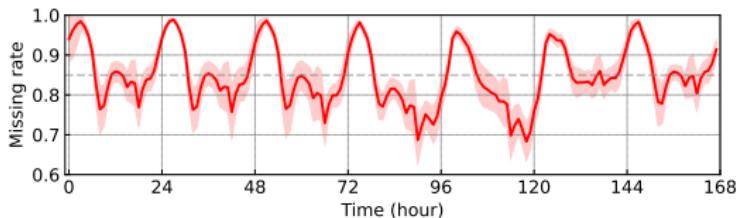
Uber Movement Data

High-dimensionality & Sparsity

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

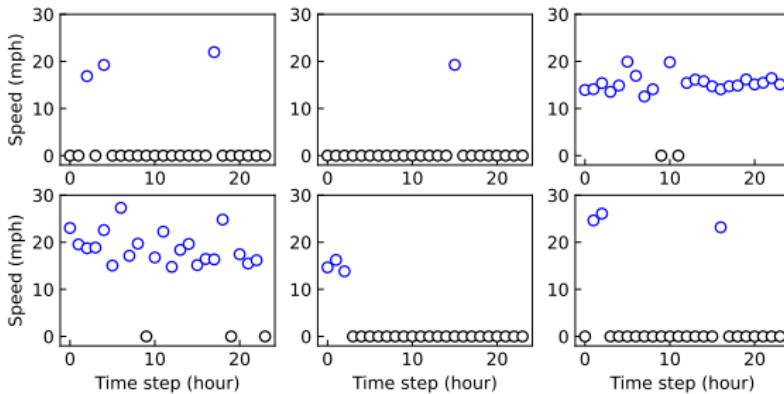


- **Seattle** movement speed data (2019)
 - **63,490** road segments & 8,760 time steps (hours)
 - Whole missing rate: **84.95%**



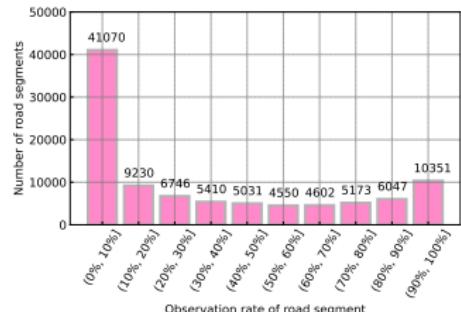
Uber Movement Data

- [Examples] Movement speed of 6 road segments on Jan. 1, 2019 in NYC.



- [Stats] Missing rate w.r.t. road segments:

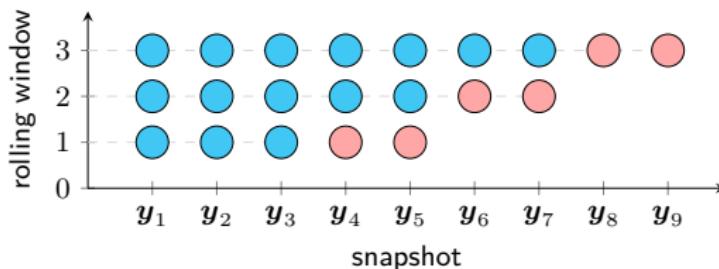
- $\approx 42\%$ road segments are with more than 90% missing values.
- $\approx 51\%$ road segments are with more than 80% missing values.
- $\approx 11\%$ road segments are with less than 10% missing values.



Experiments

Experiment setup

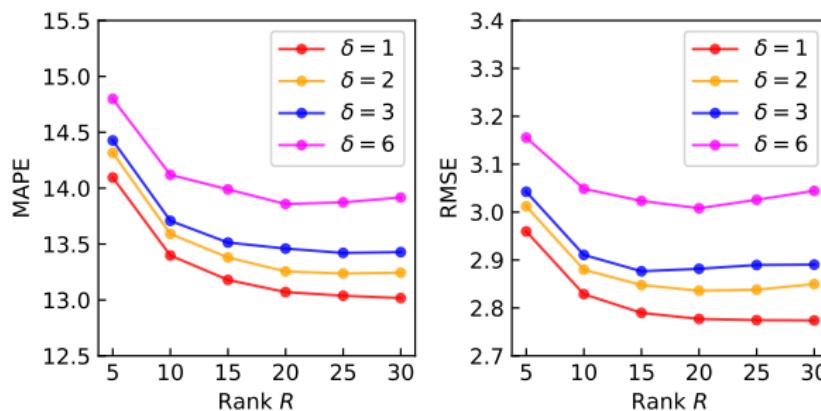
- NYC movement speed dataset:
 - Ten-week data of size 98210×1680
 - Contain 66.56% missing values
- Rolling forecasting setup:
 - Training set: 8-week data
 - Validation set: 1-week data
 - Test set: 1-week data
 - Time horizon: $\delta = 1, 2, 3, 6$
- Rolling forecasting illustration ($\delta = 2$):



Experiments

Rank selection

- A larger rank essentially provides higher accuracy but the increase becomes marginal when $R > 15$.
- We choose $R = 10$ for a good balance between performance and computational cost.

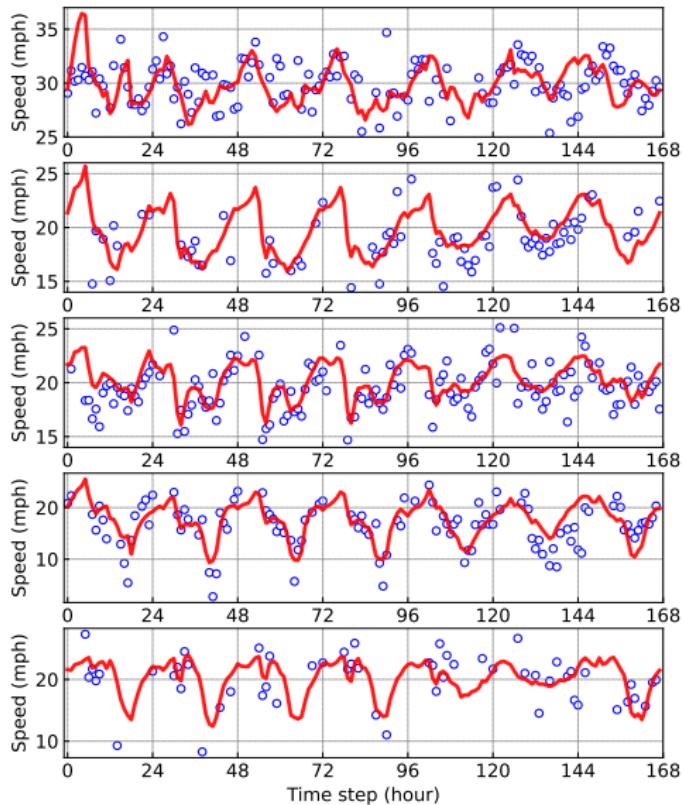


Experiments

Findings (MAPE/RMSE):

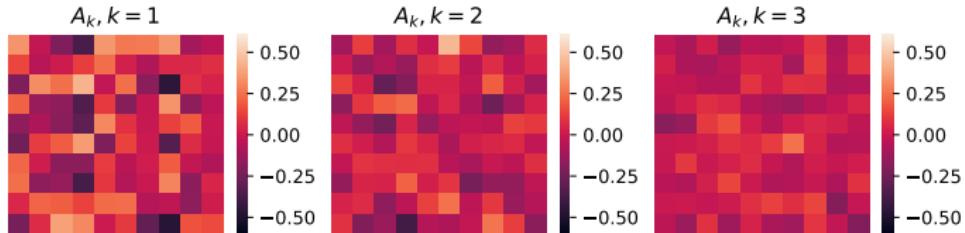
- NoTMF outperforms other models.
- Seasonal differencing in NoTMF is important.

δ	d	NoTMF ($m = 24$)	NoTMF ($m = 168$)	NoTMF-twice ($m = 168$)	TMF	TRMF	BTMF
1	1	13.63/2.88	13.53/2.86	13.45/2.85	13.74/2.90	14.50/3.12	14.94/3.13
	2	13.47/2.84	13.41/2.84	13.42/2.84	13.53/2.85	14.14/3.05	15.70/3.41
	3	13.46/2.84	13.39/2.83	13.43/2.84	13.47/2.83	13.87/2.96	15.80/3.34
	6	13.41/2.83	13.39/2.83	13.41/2.83	13.40/2.83	14.00/2.98	15.45/3.27
2	1	13.91/2.96	13.76/2.94	13.70/2.92	14.24/3.00	15.85/3.43	15.33/3.21
	2	13.77/2.92	13.63/2.89	13.72/2.92	13.87/2.91	15.04/3.31	15.87/3.32
	3	13.72/2.91	13.61/2.89	13.73/2.92	13.81/2.89	15.25/3.36	15.69/3.33
	6	13.59/2.87	13.57/2.88	13.68/2.91	13.63/2.86	14.92/3.24	15.91/3.39
3	1	14.30/3.05	14.06/3.02	14.02/3.00	14.81/3.12	17.52/3.83	15.86/3.32
	2	14.01/2.98	13.84/2.94	13.96/2.98	14.26/2.98	17.32/4.00	16.30/3.40
	3	13.95/2.97	13.79/2.93	13.98/2.98	14.04/2.94	16.91/3.71	16.56/3.49
	6	13.78/2.92	13.73/2.92	13.91/2.96	13.94/2.92	16.72/3.65	15.49/3.27
6	1	14.61/3.11	14.67/3.20	14.98/3.32	15.41/3.21	21.20/4.70	15.99/3.32
	2	14.30/3.03	14.33/3.09	14.90/3.28	14.85/3.07	20.87/5.01	16.04/3.33
	3	14.26/3.03	14.28/3.09	14.86/3.26	14.57/3.01	20.08/4.65	15.67/3.28
	6	14.06/2.97	14.16/3.06	14.80/3.23	14.47/3.00	20.40/4.35	16.38/3.50

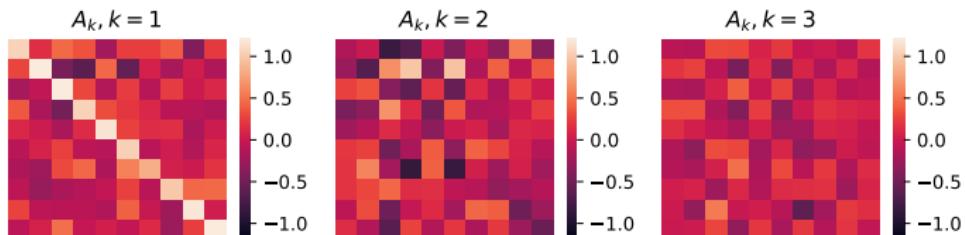


Observing A

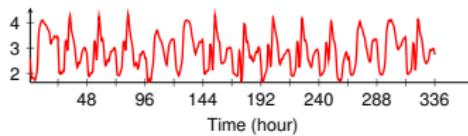
- NoTMF: showing weak correlations after differencing.



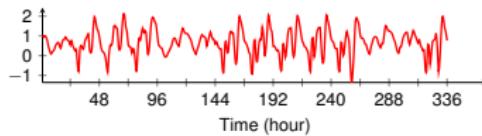
- TMF: showing strong autocorrelations in A_1 (see diagonal entries).



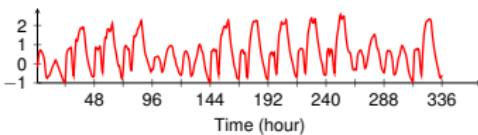
Temporal factor #1



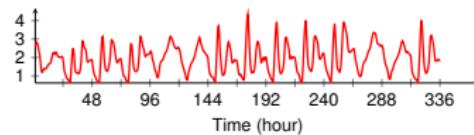
Temporal factor #2



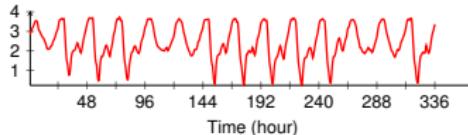
Temporal factor #3



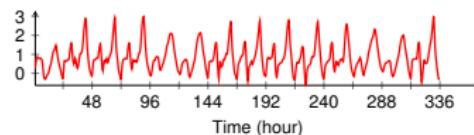
Temporal factor #4



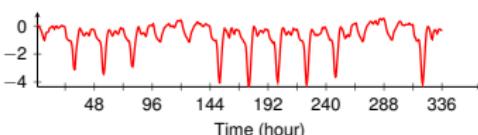
Temporal factor #5



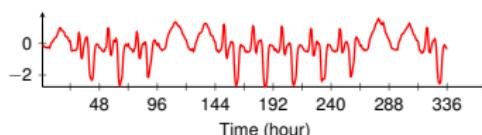
Temporal factor #6



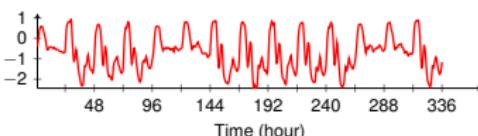
Temporal factor #7



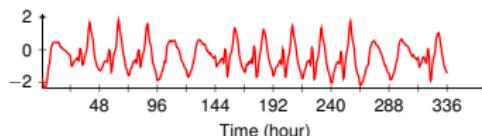
Temporal factor #8



Temporal factor #9



Temporal factor #10



Concluding Remarks

- Nonstationarity issue is important but mostly ignored.
 - e.g. the importance of stationarity in traffic forecasting (Rodrigues'22).
- Proper differencing operations are needed to address nonstationarity in real-world time series.

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

References

A short list:

- [Chen & Sun'21] X. Chen and L. Sun (2021). Bayesian temporal factorization for multidimensional time series prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [Gultekin & Paisley'18] S. Gultekin and J. Paisley (2018). Online forecasting matrix factorization. *IEEE Transactions on Signal Processing*, 67(5): 1223-1236.
- [Koren et al.'09] Y. Koren, R. Bell, and C. Volinsky (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.
- [Rao et al.'15] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon (2015). Collaborative filtering with graph information: Consistency and scalable methods. *Advances in neural information processing systems (NIPS)*.
- [Rodrigues'22] Rodrigues (2022). On the importance of stationarity, strong baselines and benchmarks in transport prediction problems. *arXiv preprint arXiv:2203.02954*.
- [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?

About me:

-  Homepage: <https://xinychen.github.io>
-  GitHub: <https://github.com/xinychen> (2.4K+ stars)
-  Blog: <https://medium.com/@xinyu.chen> (30K+ views)
-  How to reach me: chenxy346@gmail.com