

POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE



IVADO

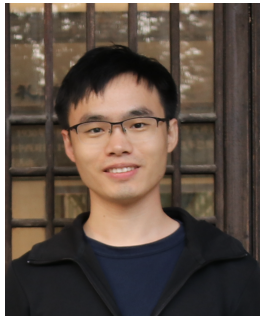
Nonstationary Temporal Matrix Factorization for Multivariate Time Series Forecasting

Xinyu Chen (Ph.D. candidate)

Polytechnique Montréal & Université de Montréal

May 11, 2022

Released on April 6, 2022.



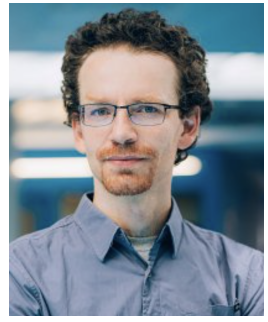
Xinyu Chen
PolyMtl



Chengyuan Zhang
McGill



Xi-Le Zhao
UESTC



Nicolas Saunier
PolyMtl



Lijun Sun
McGill

Source

1. **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.
2. **Data & Python code**: https://github.com/xinychen/tracebase
3. **Blog post**: https://medium.com/@xinyu.chen
4. **Slides**: https://xinychen.github.io/slides/notmf.pdf

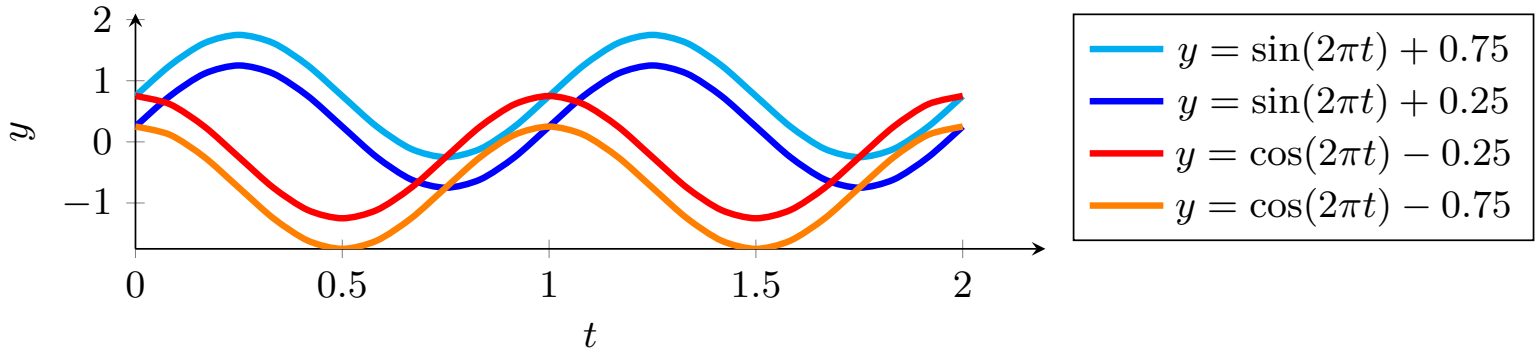
Outline

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

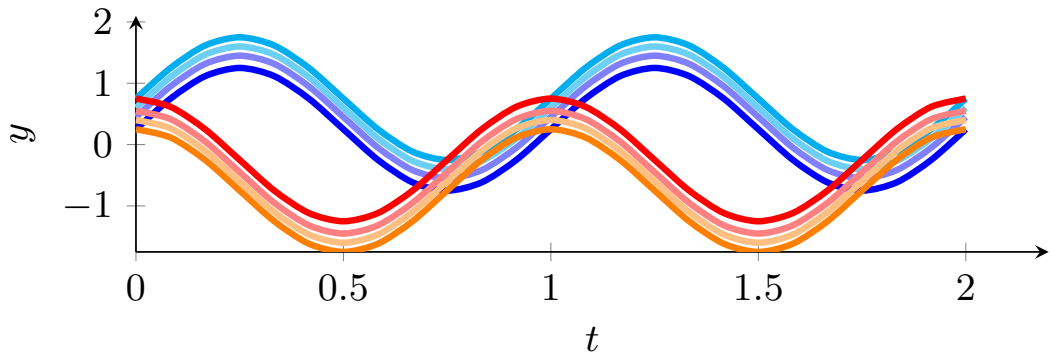
Motivation

[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.

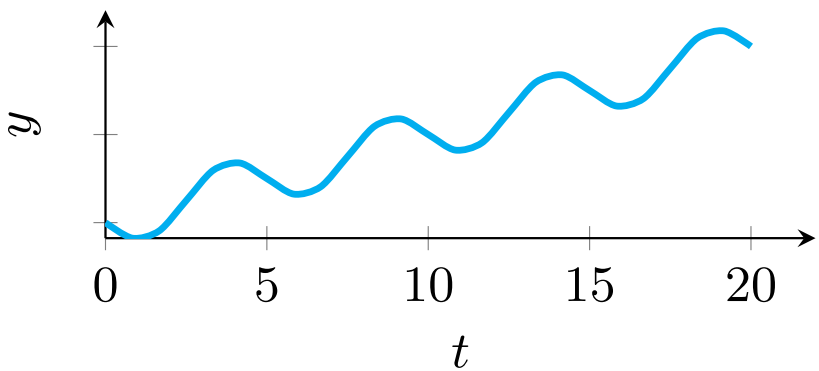


Motivation

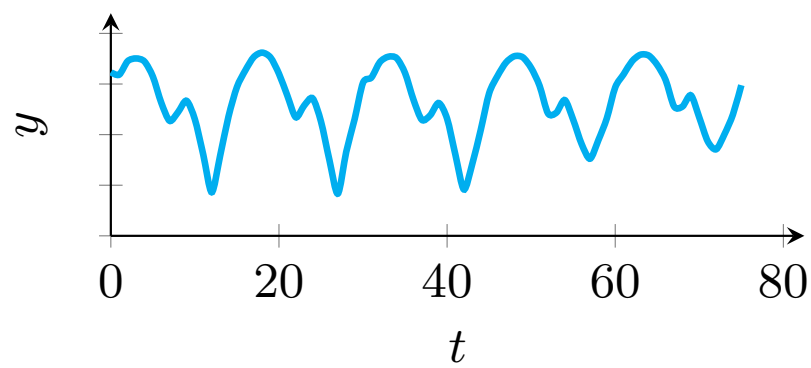
[M2] Sparse time series may reveal low-rank patterns.

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

Trends & changing levels



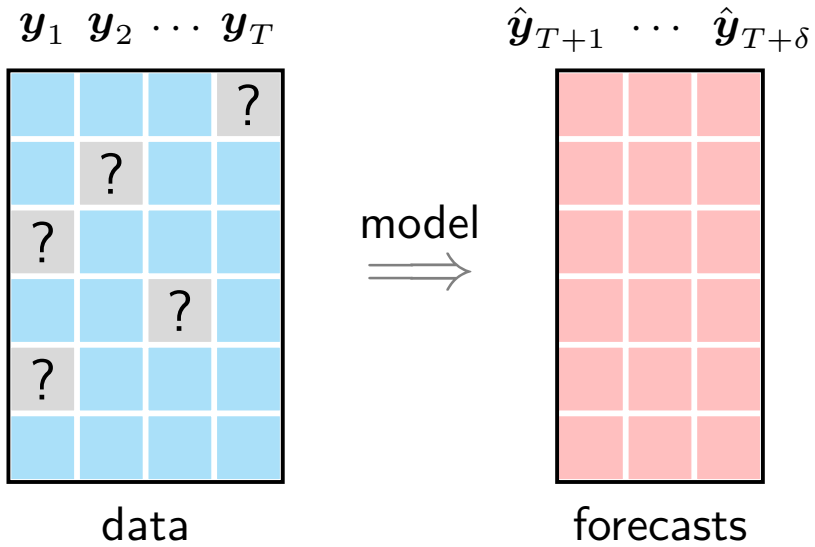
Seasonality



Objective

Multivariate time series forecasting

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



[Q]

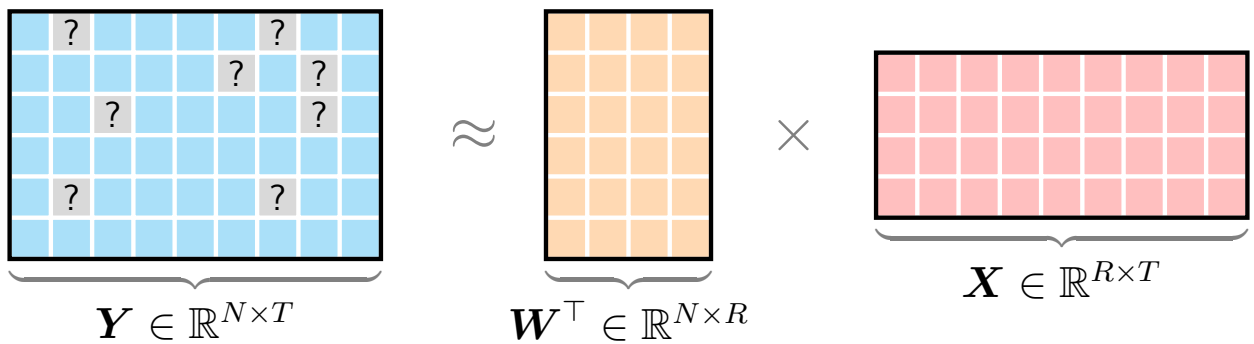
- How to learn from *high-dimensional* and *sparse* data?
- How to model *nonstationarity* in time series?
- How to perform forecasting on these time series?

Methodology

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

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

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



☹ Cannot capture temporal correlations.

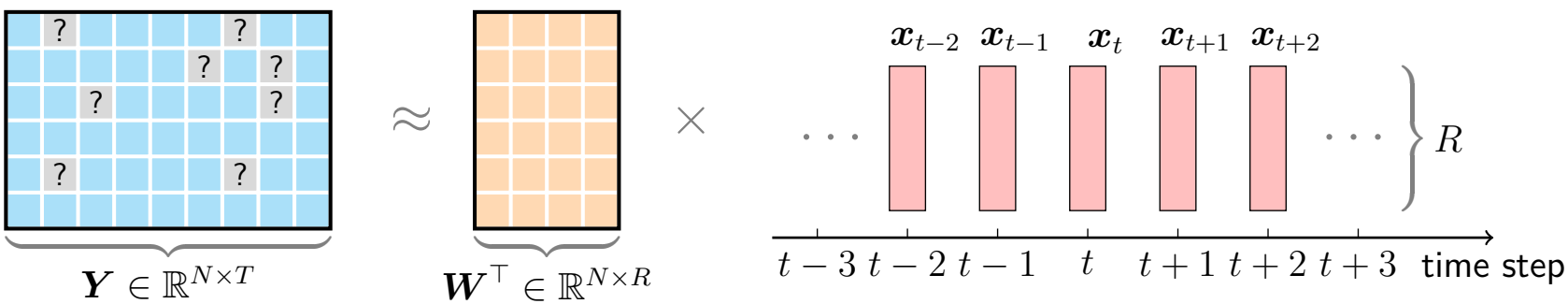
☹ Cannot perform time series forecasting.

Methodology

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

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

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



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

Methodology

Nonstationary temporal matrix factorization (NoTMF)

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

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

- First-order differencing $\boldsymbol{x}_t' = \boldsymbol{x}_t - \boldsymbol{x}_{t-1}$.
- Second-order differencing $\boldsymbol{x}_t'' = (\boldsymbol{x}_t - \boldsymbol{x}_{t-1}) - (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2})$.
- Twice-differenced series $\boldsymbol{x}_t'' = (\boldsymbol{x}_t - \boldsymbol{x}_{t-m}) - (\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-m-1})$.

🙂 Stationarizing a time series with differencing can improve the prediction.[1]

[1] Stationarity and differencing: `https://otexts.com/fpp2/stationarity.html`

Methodology

Rewrite VAR in the form of matrix

Temporal operators

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

$$\begin{aligned} \boldsymbol{\Psi}_k \triangleq & \begin{bmatrix} \boldsymbol{0}_{(T-d-m)\times(d-k)} & -\boldsymbol{I}_{T-d-m} & \boldsymbol{0}_{(T-d-m)\times(k+m)} \end{bmatrix} \\ & + \begin{bmatrix} \boldsymbol{0}_{(T-d-m)\times(d+m-k)} & \boldsymbol{I}_{T-d-m} & \boldsymbol{0}_{(T-d-m)\times k} \end{bmatrix} \\ & \in \mathbb{R}^{(T-d-m)\times T},\ k = 0, 1, \ldots, d \end{aligned} \tag{4}$$

then

$$\begin{aligned} & \sum_{t=d+m+1}^{T} \left\| (\boldsymbol{x}_t - \boldsymbol{x}_{t-m}) - \sum_{k=1}^{d} \boldsymbol{A}_k (\boldsymbol{x}_{t-k} - \boldsymbol{x}_{t-m-k}) \right\|_2^2 \\ \equiv & \left\| \boldsymbol{X}\boldsymbol{\Psi}_0^\top - \sum_{k=1}^{d} \boldsymbol{A}_k \boldsymbol{X}\boldsymbol{\Psi}_k^\top \right\|_F^2 \triangleq \left\| \boldsymbol{X}\boldsymbol{\Psi}_0^\top - \boldsymbol{A}(\boldsymbol{I}_d \otimes \boldsymbol{X})\boldsymbol{\Psi}^\top \right\|_F^2 \end{aligned} \tag{5}$$

where $\boldsymbol{A} \triangleq \begin{bmatrix} \boldsymbol{A}_1 & \cdots & \boldsymbol{A}_d \end{bmatrix}$ and $\boldsymbol{\Psi} \triangleq \begin{bmatrix} \boldsymbol{\Psi}_1 & \cdots & \boldsymbol{\Psi}_d \end{bmatrix}$.

Methodology

Rewrite NoTMF:

$$\begin{aligned}\min_{\boldsymbol{W},\boldsymbol{X},\boldsymbol{A}} \ & \frac{1}{2}\|\mathcal{P}_{\Omega}(\boldsymbol{Y}-\boldsymbol{W}^{\top}\boldsymbol{X})\|_F^2+\frac{\rho}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{X}\|_F^2) \\ & +\frac{\lambda}{2}\|\boldsymbol{X}\boldsymbol{\Psi}_0^{\top}-\boldsymbol{A}(\boldsymbol{I}_d\otimes\boldsymbol{X})\boldsymbol{\Psi}^{\top}\|_F^2\end{aligned} \tag{6}$$

Alternating minimization method:

- w.r.t. $\boldsymbol{W}$:

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

- w.r.t. $\boldsymbol{X}$:

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

- w.r.t. $\boldsymbol{A}$:

$$\boldsymbol{A}=\boldsymbol{X}\boldsymbol{\Psi}_0^{\top}[(\boldsymbol{I}_d\otimes\boldsymbol{X})\boldsymbol{\Psi}^{\top}]^{\dagger}$$

Methodology

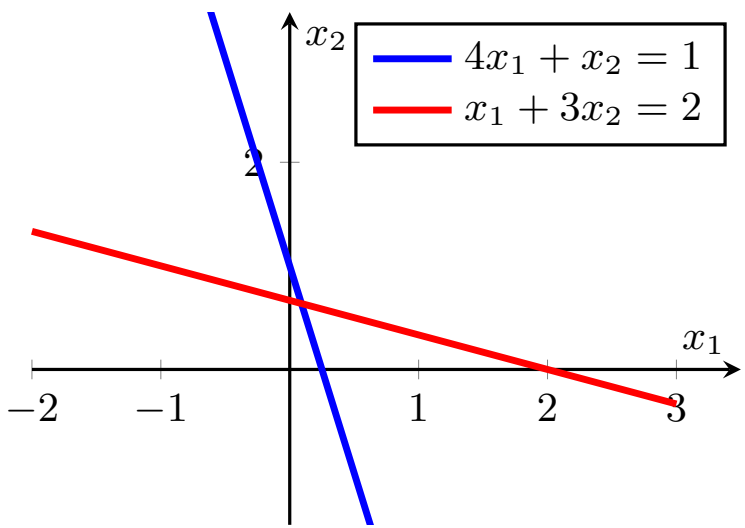
Solve generalized Sylvester equation (w.r.t. $\boldsymbol{X}$):

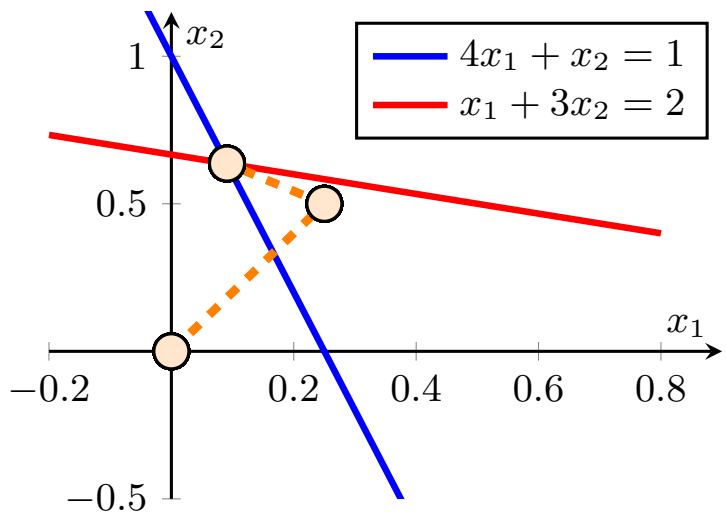
$$\underbrace{\boldsymbol{W}\mathcal{P}_\Omega(\boldsymbol{W}^\top\boldsymbol{X})+\rho\boldsymbol{X}+\lambda\sum_{k=0}^{d}\boldsymbol{A}_k^\top\left(\sum_{h=0}^{d}\boldsymbol{A}_h\boldsymbol{X}\boldsymbol{\Psi}_h^\top\right)\boldsymbol{\Psi}_k}_{\mathcal{L}_x(\boldsymbol{X})\triangleq\mathsf{vec}(\cdot)}=\boldsymbol{W}\mathcal{P}_\Omega(\boldsymbol{Y})$$

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

Methodology

- **[Q]** $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with known $\boldsymbol{A} \in \mathbb{R}^{n\times n}, \boldsymbol{b} \in \mathbb{R}^n$, the solution of $\boldsymbol{x} \in \mathbb{R}^n$?
- **Conjugate gradient** method:
  - Initialize $\boldsymbol{x}$ as $\boldsymbol{x}_0$, and compute the residual $r_0 := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$. Let $\boldsymbol{q}_0 := \boldsymbol{r}_0$.
  - Repeat:
    - Compute $\alpha_\ell := \frac{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}{\boldsymbol{q}_\ell^\top \boldsymbol{A}\boldsymbol{q}_\ell}$, and update $\begin{cases} \boldsymbol{x}_{\ell+1} := \boldsymbol{x}_\ell + \alpha_\ell \boldsymbol{q}_\ell \\ \boldsymbol{r}_{\ell+1} := \boldsymbol{r}_\ell - \alpha_\ell \boldsymbol{A}\boldsymbol{q}_\ell \end{cases}$.
    - Compute $\beta_\ell := \frac{\boldsymbol{r}_{\ell+1}^\top \boldsymbol{r}_{\ell+1}}{\boldsymbol{r}_\ell^\top \boldsymbol{r}_\ell}$, and update $\boldsymbol{q}_{\ell+1} := \boldsymbol{r}_{\ell+1} + \beta_\ell \boldsymbol{q}_\ell$.
    - Update $\ell := \ell + 1$.

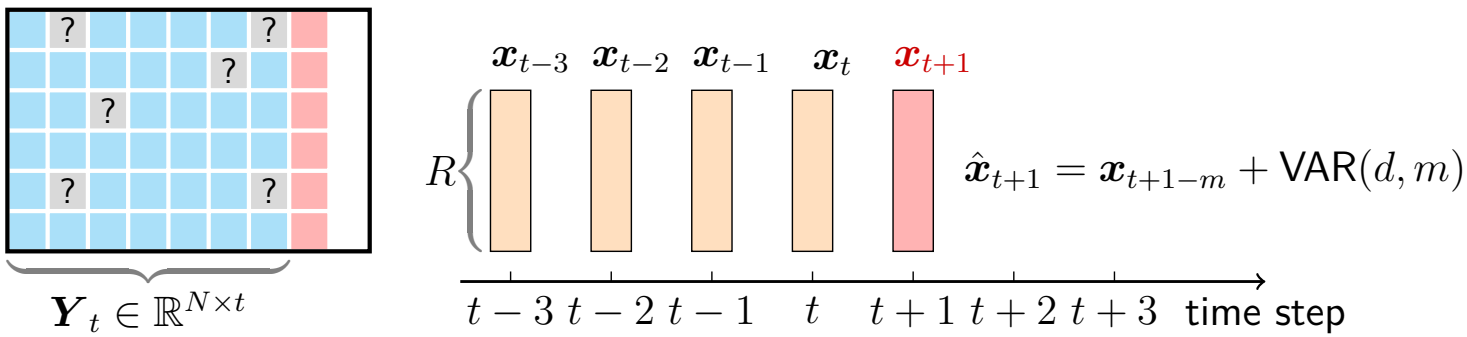




Methodology

NoTMF forecasting on streaming data?

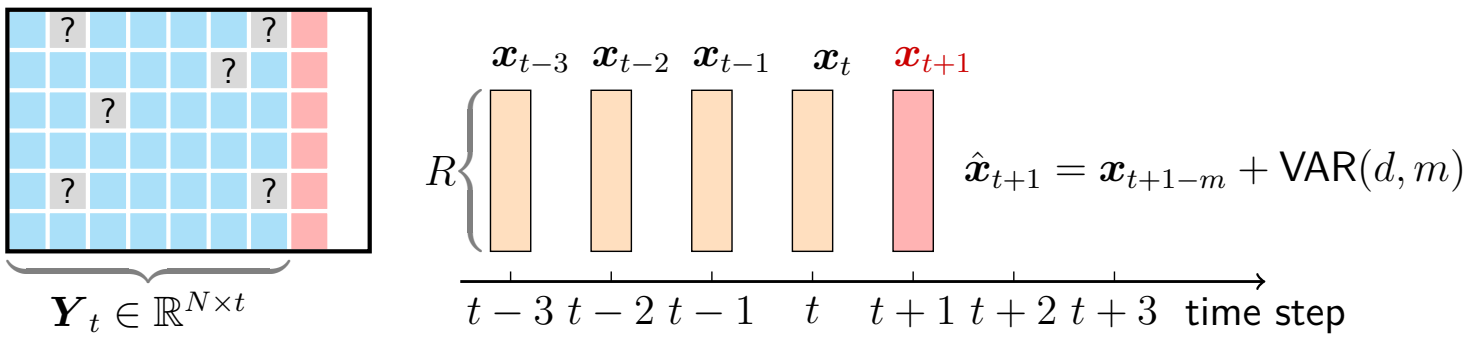
- NoTMF: Use $\boldsymbol{Y}_t$ to estimate $\{\boldsymbol{W}, \boldsymbol{X}, \boldsymbol{A}\}$.



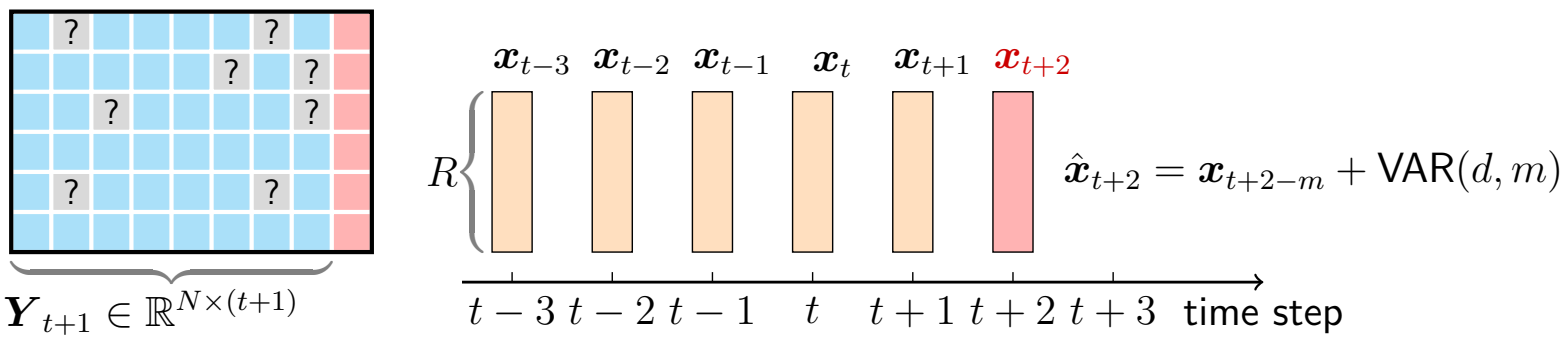
Methodology

NoTMF forecasting on streaming data?

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



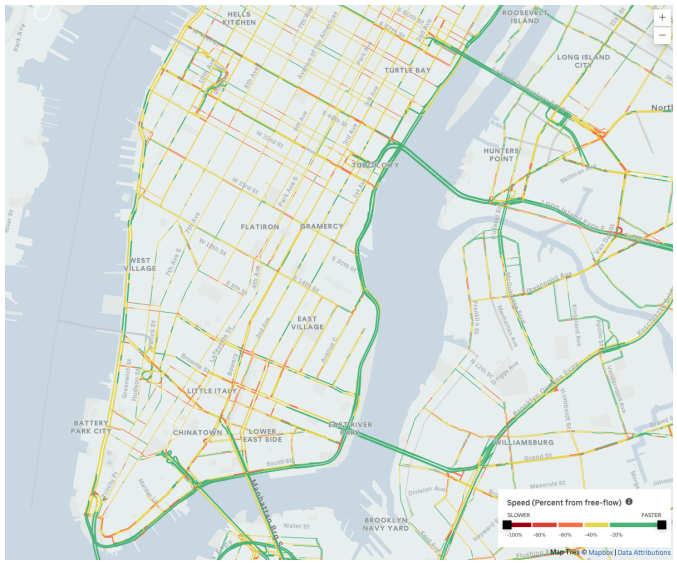
- Online forecasting (Gultekin & Paisley'18): Fix $\boldsymbol{W}$ and use $\boldsymbol{Y}_{t+1}$ to update $\{\boldsymbol{X}, \boldsymbol{A}\}$.



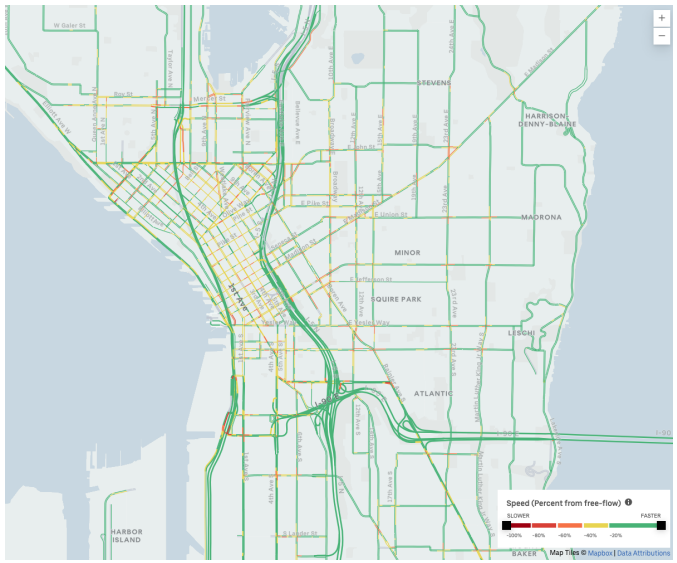
Uber Movement Data

High-dimensionality & Sparsity

- Uber (hourly) movement speed data[2]



NYC movement



Seattle movement

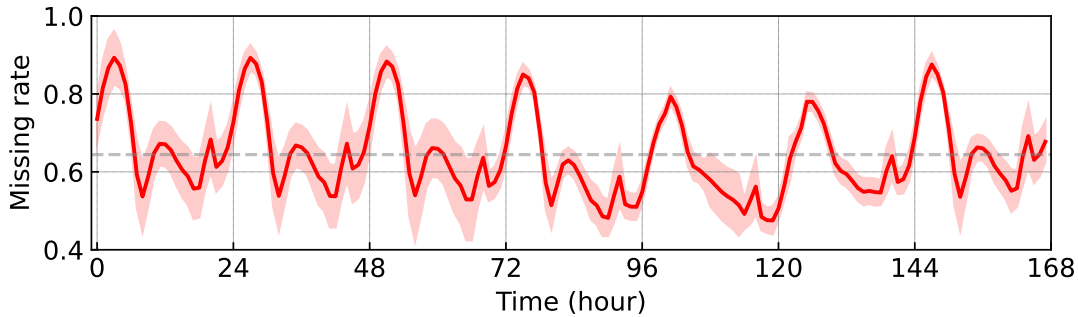
- The average speed on a given road segment for each hour of each day.
- Hourly speeds are computed when road segments have 5+ unique trips.
- **Issue**: insufficient sampling of ridesharing vehicles on the road network.

[2] `https://movement.uber.com/`

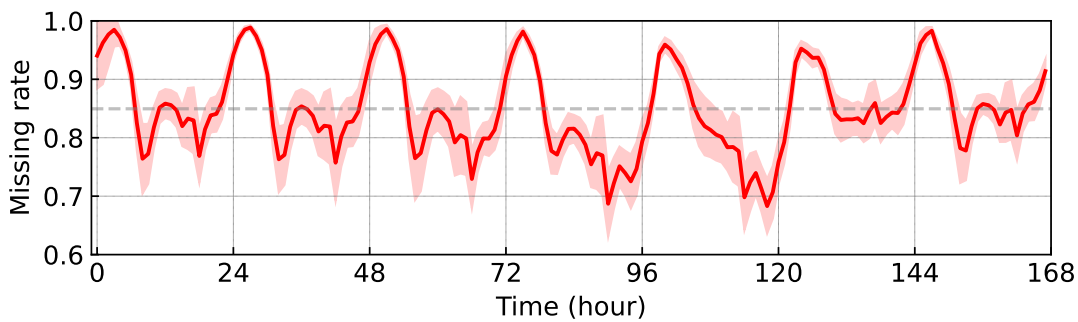
Uber Movement Data

High-dimensionality & Sparsity

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

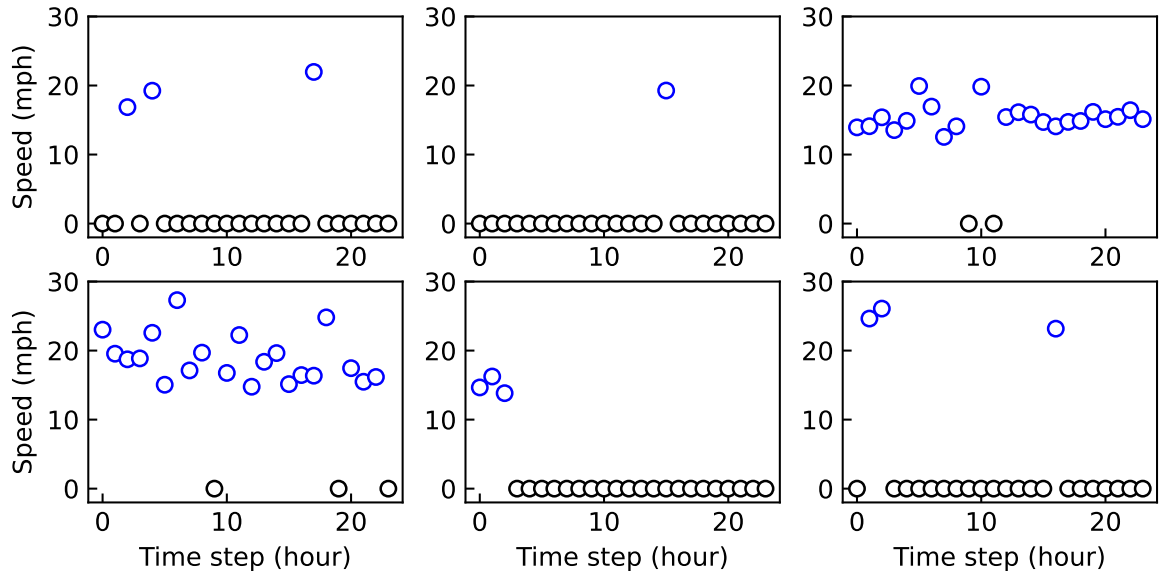


- **Seattle** movement speed data (2019)
  - **63,490** road segments & 8,760 time steps (hours)
  - Whole missing rate: **84.95%**

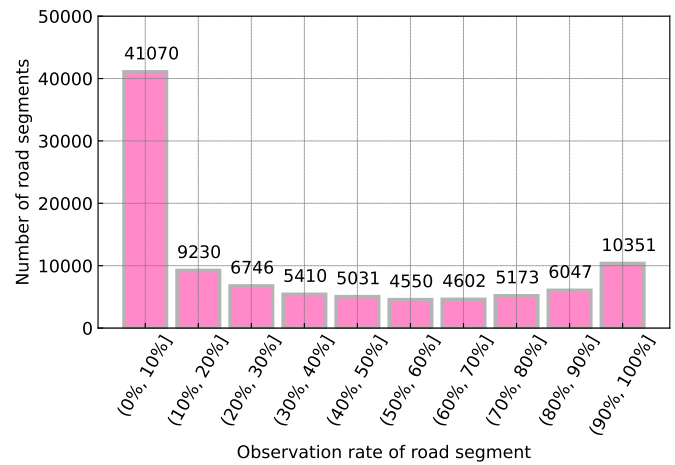


Uber Movement Data

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



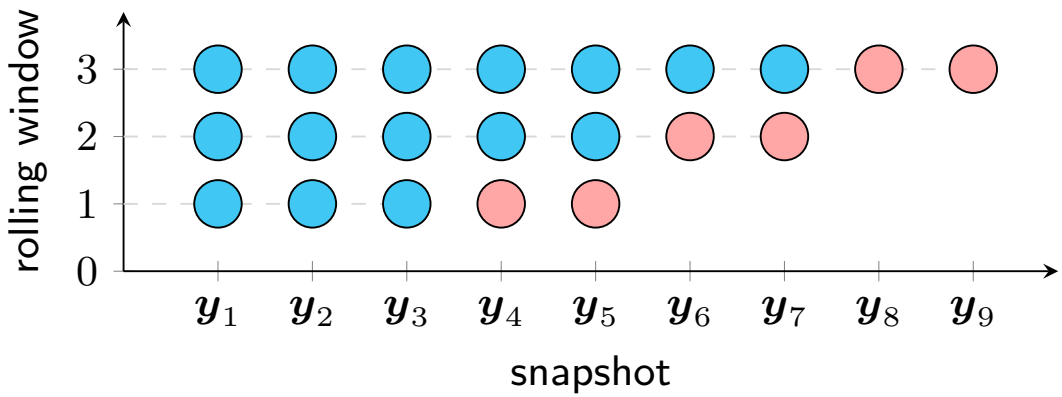
- **[Stats]** Missing rate w.r.t. road segments:
  - $\approx 42\%$ road segments are with more than 90% missing values.
  - $\approx 51\%$ road segments are with more than 80% missing values.
  - $\approx 11\%$ road segments are with less than 10% missing values.



Experiments

Experiment setup

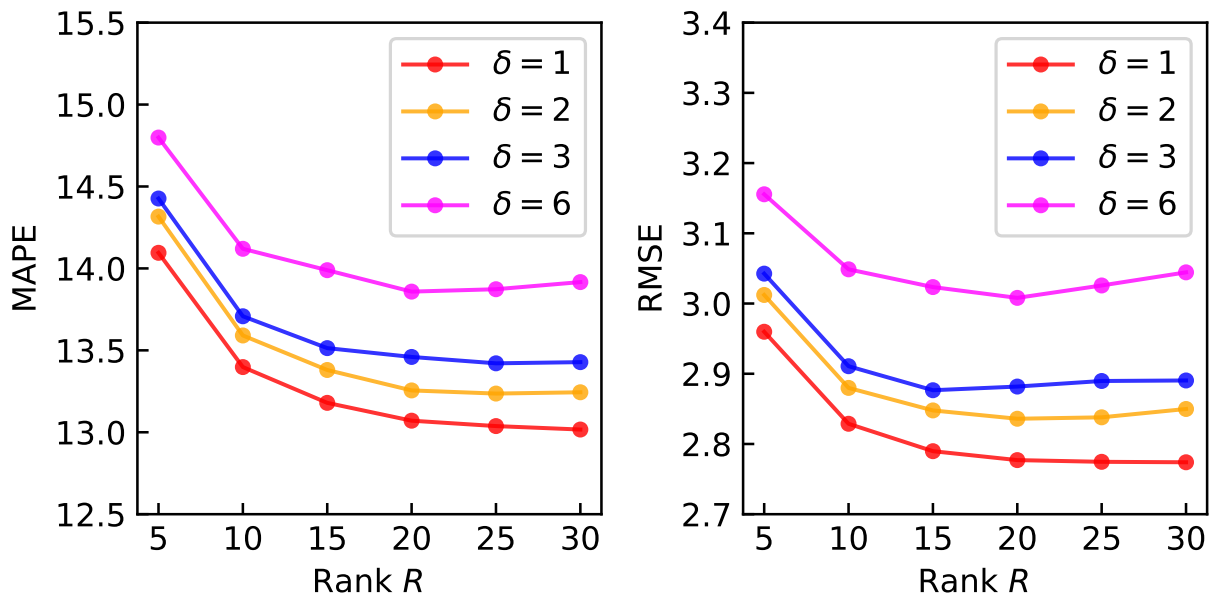
- NYC movement speed dataset:
  - Ten-week data of size 98210×1680
  - Contain 66.56% missing values
- Rolling forecasting setup:
  - Training set: 8-week data
  - Validation set: 1-week data
  - Test set: 1-week data
  - Time horizon: $\delta = 1, 2, 3, 6$
- Rolling forecasting illustration ($\delta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



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.

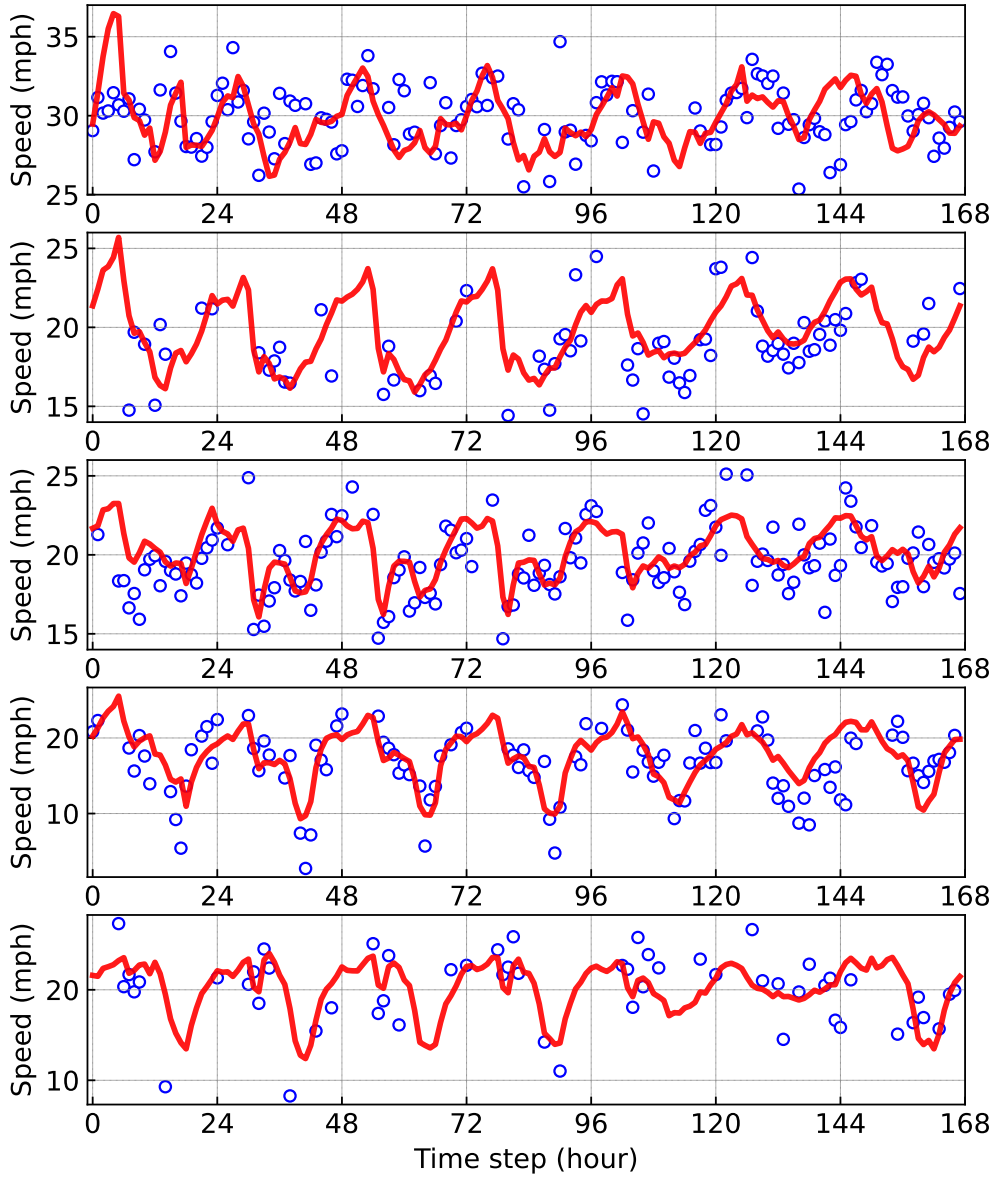


Experiments

Findings (MAPE/RMSE):

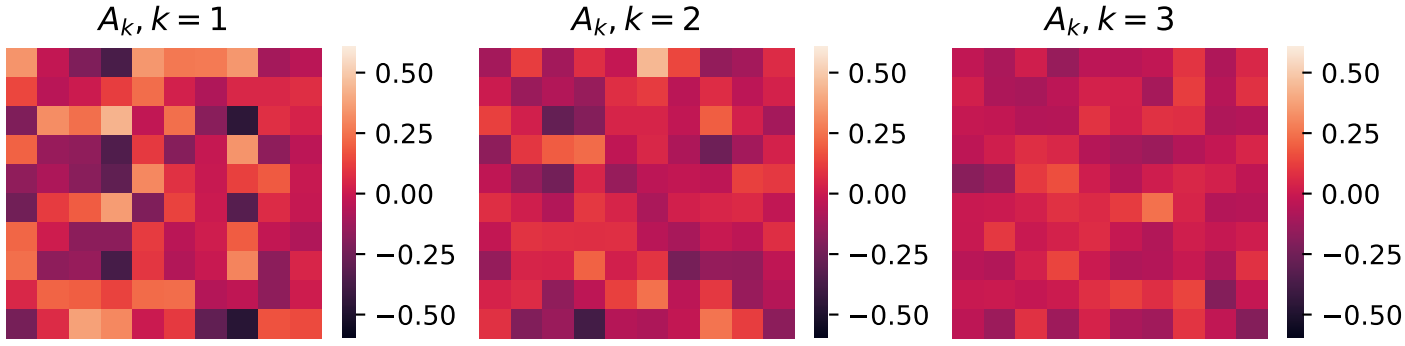
- NoTMF outperforms other models.
- Seasonal differencing in NoTMF is important.

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

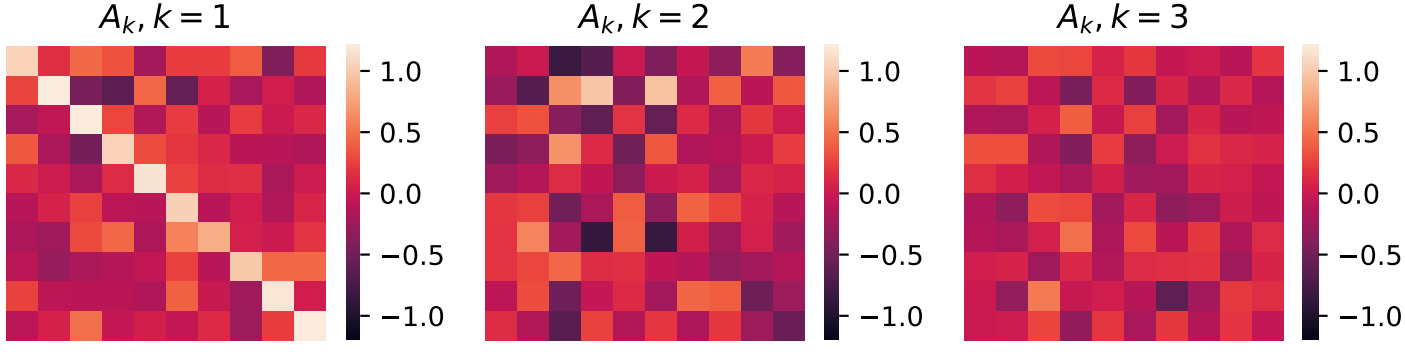

Speed (mph)
35
30
25
0
24
48
72
96
120
144
168
Speed (mph)
25
20
15
0
24
48
72
96
120
144
168
Speed (mph)
25
20
15
0
24
48
72
96
120
144
168
Speed (mph)
20
10
0
24
48
72
96
120
144
168
Speed (mph)
20
10
0
24
48
72
96
120
144
168
Time step (hour)

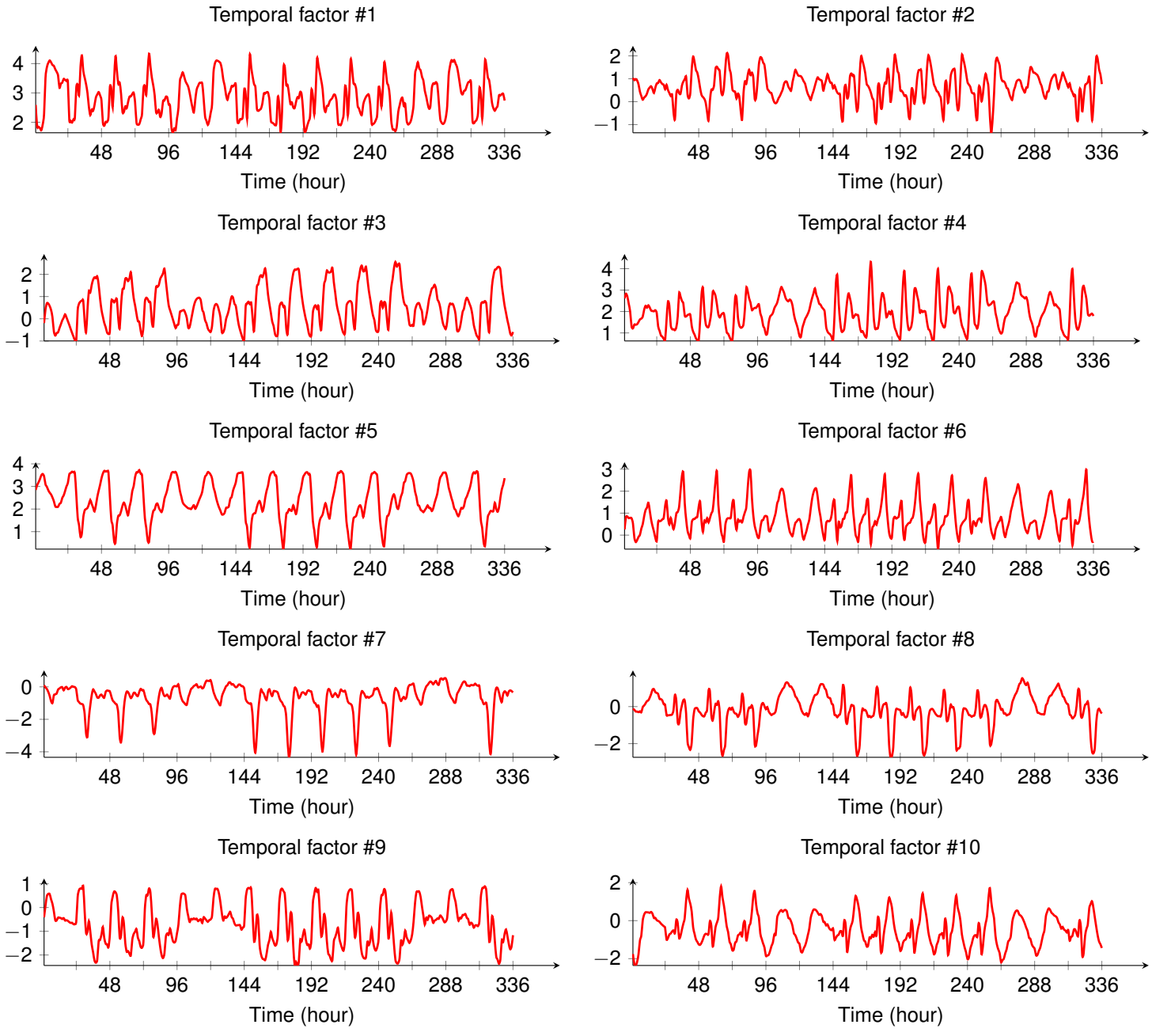
Observing A

- NoTMF: showing weak correlations after differencing.



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




Temporal factor #1
Time (hour)
Temporal factor #2
Time (hour)
Temporal factor #3
Time (hour)
Temporal factor #4
Time (hour)
Temporal factor #5
Time (hour)
Temporal factor #6
Time (hour)
Temporal factor #7
Time (hour)
Temporal factor #8
Time (hour)
Temporal factor #9
Time (hour)
Temporal factor #10
Time (hour)

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).



POLYTECHNIQUE MONTRÉAL

UNIVERSITÉ D'INGÉNIERIE



IVADO

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