



Open-Source Projects: Machine Learning for Transportation Data Imputation and Prediction

 $\label{eq:Research Workshop}$ TRB 103rd Annual Meeting \cdot Washington, D.C., USA

Xinyu Chen

January 11, 2024

Open-source & reproducible research:

• GitHub: https://github.com/xinychen

② Slides: https://xinychen.github.io/slides/transdim.pdf

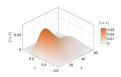
Project website: https://spatiotemporal-data.github.io

$\mathsf{ML}\ \mathsf{algorithms}$



transdim
(1.1k stars)

Visualization tools



1. Storytelling with Data

- 2. Spatiotemporal Traffic Data Modeling
 - Reformulate traffic data imputation
 - Reformulate traffic forecasting
- 3. Python Implementation
 - Traffic data processing
 - Switch from CPU to GPU
- 4. "Sustainable" Research
 - Post something that matters

Storytelling with Data

• Uber (hourly) movement speed data



NYC movement

Seattle movement

- {road segment, time step (hour), average speed}
- ullet $Y \in \mathbb{R}^{N imes T}$ with N spatial locations imes T time steps
- Computing hourly speed: Road segments have 5+ unique trips.

Issue: Insufficient sampling of ridesharing vehicles on the road network!

Storytelling with Data

- Data
- Quality
- Sparsity
- Estimation
- Imputation
- Interpolation
- Forecasting

Storytelling with Data

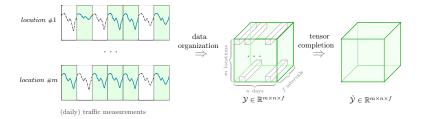
Reformulate Traffic Data Imputation

Imputing missing traffic data

Represent traffic data as tensors

Tensorization:
$$Y \in \mathbb{R}^{m \times t} \to \mathcal{Y} \in \mathbb{R}^{m \times n \times f}$$

w/m locations, n days, and f time intervals per day.



• Tensor completion (Observed index set Ω)



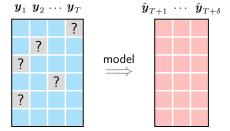
Reformulate Traffic Data Imputation

Reformulate Traffic Forecasting

Forecasting urban traffic states with sparse data

• Problem definition (δ -step ahead forecasting)

$$\underbrace{\{\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_T\}}_{\text{Current traffic states}} \qquad \underbrace{\{\hat{\boldsymbol{y}}_{T+1},\hat{\boldsymbol{y}}_{T+2},\ldots,\hat{\boldsymbol{y}}_{T+\delta}\}}_{\text{Future traffic states}}$$



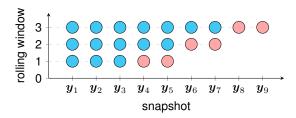
Reformulate Traffic Forecasting

(Rolling) Forecasting urban traffic states with sparse data

1st rolling step: $\{oldsymbol{y}_1, oldsymbol{y}_2, oldsymbol{y}_3\}
ightarrow \{oldsymbol{y}_4, oldsymbol{y}_5\}$

2nd rolling step: $\{{m y}_1,{m y}_2,{m y}_3,{m y}_4,{m y}_5\} o \{{m y}_6,{m y}_7\}$

3rd rolling step: $\{{m y}_1,{m y}_2,{m y}_3,{m y}_4,{m y}_5,{m y}_6,{m y}_7\} o \{{m y}_8,{m y}_9\}$



TMF

TMF^{1,2} Jupyter Notebook

¹tracebase: https://github.com/xinychen/tracebase

²tpami

Python Implementation

- Jupyter Notebook
- NumPy/SciPy: computing, Pandas: data processing

Traffic Data Processing

Computing with numpy (numerical computing in Python)

- Data format: .npz (compressed format of .npy)
- Example
- Easy to connect with numpy (in CPU environment) & cupy (in GPU environment)

NYC Uber Movement

In this repository, we prepare the dataset and place it at the folder datasets/NYC-movement-data-set:

- hourly_speed_mat_2019_1.npz (91 MB): data is of size 98,210 x 744 with 23,228,581 positive speed observations.
- hourly_speed_mat_2019_2.npz (85.2 MB): data is of size 98,210 x 672 with 21,912,460 positive speed observations.
- hourly_speed_mat_2019_3.npz (38.1 MB): data is of size 98,210 x 264 with 10,026,045 positive speed observations.

Switch from CPU to GPU

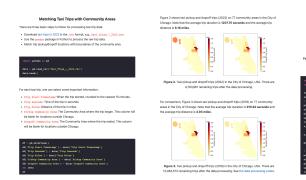
Python implementation of algorithms with the numpy package (Using less packages can improve the reproducibility) Easy to convert the codes from CPU to GPU

import numpy as np \Rightarrow import cupy as np

Post Something That Matters

Post well-documented data processing files (e.g., processing Chicago taxi data)

- Beginners to build coding skills
- Researchers to build research ideas





Source: https://spatiotemporal-data.github.io/Chicago-mobility/taxi-data

Post Something That Matters

Post scientific problems (e.g., spatiotemporal data modeling)

Optimizing Interpretable Time-Varying Autoregression with Orthogonal Constraints

Generally speaking, any spatiotemporal data in the form of a matrix can be written as $Y \in \mathbb{R}^{N \times T}$ with N spatial areas/nectations and T time steps. To discover interpretable spatial/temporal patterns, one can build a time-varying autoregression on the time snapshots $y_1, y_2, \ldots, y_T \in \mathbb{R}^N$ (Chen et al., 2023). The time-varying coefficients in the autoregression allow one to characterize the time-varying system behavior, but the challenges still remain.

To capture interpretable modes/patterns, one can use tensor factorization formulas to parameterize the coefficients and the optimization problem can be easily built. However, a great challenge would be how to make the modes "more interpretable", specifically, e.g., how to learn orthogonal modes in the modeling process. In this post, we present an optimization problem of the time-varying autoregression with orthogonal constraints as follows.

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{G},\boldsymbol{V},\boldsymbol{X}} & \frac{1}{2} \sum_{t=2}^{T} \left\| \boldsymbol{y}_{t} - \boldsymbol{W} \boldsymbol{G}(\boldsymbol{x}_{t}^{\top} \otimes \boldsymbol{V})^{\top} \boldsymbol{y}_{t-1} \right\|_{2}^{2} \\ \text{s.t.} & \begin{cases} \boldsymbol{W}^{\top} \boldsymbol{W} = \boldsymbol{I}_{R} \\ \boldsymbol{V}^{\top} \boldsymbol{V} = \boldsymbol{I}_{R} \\ \boldsymbol{X}^{\top} \boldsymbol{X} = \boldsymbol{I}_{R} \end{cases} \end{aligned}$$

where $W \in \mathbb{R}^{N \times R}$ and $X \in \mathbb{R}^{(T-1) \times R}$ refer to as the spatial modes and the temporal modes, respectively. This model can discover urban mobility transition patterns.

Source: https://spatiotemporal-data.github.io/probs/orth-var

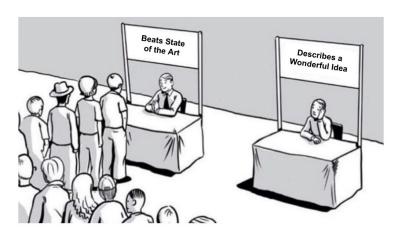
Why?

Academic:

- Sustainable research environment (w.r.t. our team & followers)
- Interact with researchers from different fields
- Provide platform and benchmark for comparison
- Stimulate new algorithmic ideas

Industry:

Solution to ...



Source: Twitter





Thanks for your attention!

Any Questions?

About me:

★ Homepage: https://xinychen.github.io

GitHub: https://github.com/xinychen