



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE



transdim: Machine Learning for Transportation Data Imputation and Prediction

Reproducible Research Workshop

TRB 103rd Annual Meeting · Washington, D.C., USA

Xinyu Chen

January 11, 2024

Open-source & reproducible research:

- ① GitHub: <https://github.com/xinychen>
- ② Slides: <https://xinychen.github.io/slides/transdim.pdf>

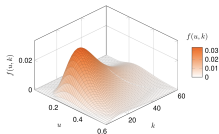
ML algorithms



transdim

(1.1k stars)

Visualization tools



awesome-latex-drawing

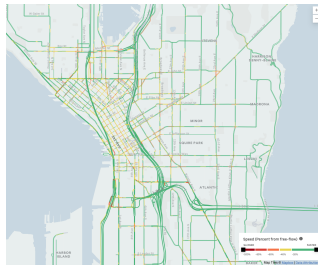
(1.2k stars)

Storytelling with Data

- Uber (hourly) movement speed data



NYC movement



Seattle movement

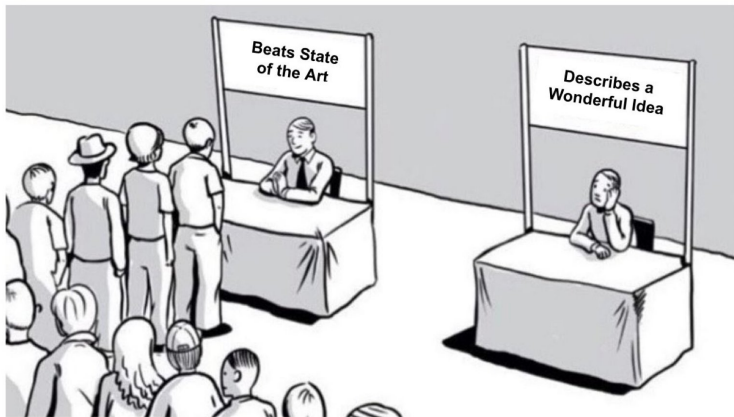
- $\{\text{road segment, time step (hour), average speed}\}$
- $\mathbf{Y} \in \mathbb{R}^{N \times T}$ with N spatial locations $\times T$ time steps
- Computing hourly speed: Road segments have 5+ unique trips.

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

Storytelling with Data

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

Storytelling with Data



Source: Twitter

Computing with `numpy` (numerical computing in Python)

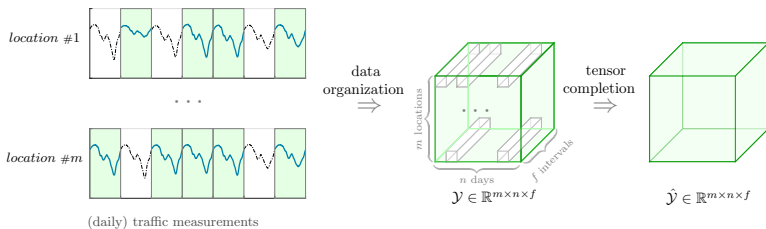
- Data format: `.npz` (compressed format of `.npy`)
- Example
- Easy to connect with `numpy` (in CPU environment) & `cupy` (in GPU environment)

Reformulate Traffic Data Imputation

- Represent traffic data as tensors

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

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



- Tensor completion (Observed index set Ω)

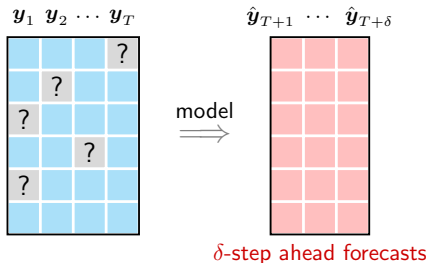
$$\underbrace{\mathcal{P}_{\Omega}(\mathcal{Y})}_{\text{Partially observed}} \xrightarrow{\text{Estimate}} \underbrace{\mathcal{P}_{\Omega}^{\perp}(\mathcal{Y})}_{\text{Unobserved}}$$

Reformulate Traffic Forecasting

Forecasting urban traffic states with sparse data

- Problem definition

$$\underbrace{\{y_1, y_2, \dots, y_T\}}_{\text{Current traffic states}} \xrightarrow{\text{Estimate}} \underbrace{\{\hat{y}_{T+1}, \hat{y}_{T+2}, \dots, \hat{y}_{T+\delta}\}}_{\text{Future traffic states}}$$



Switch from CPU to GPU

Python implementation of algorithms with the `numpy` package
Easy to convert the codes from CPU to GPU

```
import numpy as np    ⇒    import cupy as np
```

Why?

Academic:

- Provide platform and benchmark for comparison
- Stimulate new algorithmic ideas

Industry:

- Solution to ...

Next-step plan:

- Spatiotemporal data modeling:
<https://spatiotemporal-data.github.io>



**POLYTECHNIQUE
MONTREAL**

UNIVERSITÉ
D'INGÉNIERIE



IVADO


Thanks for your attention!

Any Questions?

About me:

 Homepage: <https://xinychen.github.io>

 GitHub: <https://github.com/xinychen>

 How to reach me: chenxy346@gmail.com