Professorship in Transportation Analytics (Teaching Concept & Philosophy)

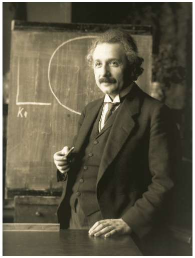
Xinyu Chen

Postdoctoral Associate, MIT

December 6, 2024

Education Mission

"Education is not the learning of facts, but the training of the mind to think."



Albert Einstein

Source: Wiki

- TUM School of Management in Heilbronn:
  - Bachelor in Management & Data Science
  - Master in Management
  - Master in Management & Digital Technology
  - Master in Management & Innovation
- My targets: Fostering curiosity and empowering students to take ownership of their learning; Creating an inclusive and engaging environment that supports diverse students.

Teaching Methods

Format & Assessment:

❶ Research-based techniques (critical thinking skills)

❷ Class quiz (problem solving)

❸ Project-oriented examination (problem solving)

❹ Mid-term/final-term presentation

Classroom is a family:

❶ Making learning fun

❷ Making it as simple and clear as possible

❷ Coffee time for discussion (guidance & support & thought exchange)

Source: link

Collaboration:

❶ Approaching advanced AI techniques

❷ Building blog posts (e.g., ICLR blogposts track)

Source: ICLR 2024

Methods for Assessing Students' Learning

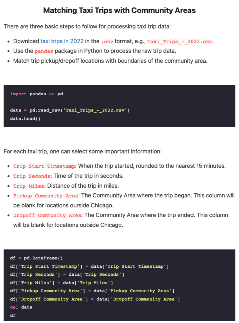
Assessment of Teaching

Data

Well-documented data files:

❶ Beginners to build coding skills

❷ Graduate students to build research projects



Matching Taxi Trips with Community Areas

There are three basic steps to follow for processing taxi trip data:

- Download taxi trips in 2022 in the `.csv` format, e.g., `Taxi_Trips_-_2022.csv`.
- Use the `pandas` package in Python to process the raw trip data.
- Match trip pickup/dropoff locations with boundaries of the community area.

```python
import pandas as pd

data = pd.read_csv('Taxi_Trips_-_2022.csv')
data.head()
```

For each taxi trip, one can select some important information:

- `Trip Start Timestamp`: When the trip started, rounded to the nearest 15 minutes.
- `Trip Seconds`: Time of the trip in seconds.
- `Trip Miles`: Distance of the trip in miles.
- `Pickup Community Area`: The Community Area where the trip began. This column will be blank for locations ourside Chicago.
- `Dropoff Community Area`: The Community Area where the trip ended. This column will be blank for locations outside Chicago.

```python
df = pd.DataFrame()
df['Trip Start Timestamp'] = data['Trip Start Timestamp']
df['Trip Seconds'] = data['Trip Seconds']
df['Trip Miles'] = data['Trip Miles']
df['Pickup Community Area'] = data['Pickup Community Area']
df['Dropoff Community Area'] = data['Dropoff Community Area']
del data
df
```

Figure 2 shows taxi pickup and dropoff trips (2022) on 77 community areas in the City of Chicago. Note that the average trip duration is **1207.75 seconds** and the average trip distance is **6.16 miles**.

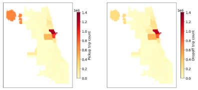


Figure 2. Taxi pickup and dropoff trips (2022) in the City of Chicago, USA. There are 4,763,961 remaining trips after the data processing.

For comparison, Figure 3 shows taxi pickup and dropoff trips (2019) on 77 community areas in the City of Chicago. Note that the average trip duration is **915.62 seconds** and the average trip distance is **3.93 miles**.

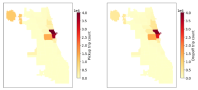


Figure 3. Taxi pickup and dropoff trips (2019) in the City of Chicago, USA. There are 12,484,572 remaining trips after the data processing. See the data processing codes.

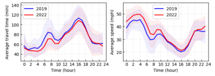
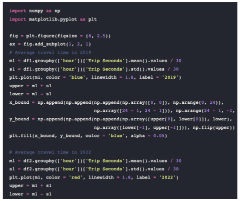


Figure 6. Average travel time and speed from area 32 (i.e., Downtown) to area 76 (i.e., Airport) in both 2019 and 2022.



```python
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure(figsize = (8, 2.5))
ax = fig.add_subplot(1, 2, 1)
# Average travel time in 2019
m1 = df1.groupby(['hour'])['Trip Seconds'].mean().values / 30
s1 = df1.groupby(['hour'])['Trip Seconds'].std().values / 30
plt.plot(m1, color = 'blue', linewidth = 1.8, label = '2019')
upper = m1 + s1
lower = m1 - s1
x_bound = np.append(np.append(np.append(np.array([0, 0]), np.arange(0, 24)),
                    np.array([24 - 1, 24 - 1])), np.arange(24 - 1, -1,
y_bound = np.append(np.append(np.append(np.array([upper[0], lower[0]]), lower),
                    np.array([lower[-1], upper[-1]])), np.flip(upper))
plt.fill(x_bound, y_bound, color = 'blue', alpha = 0.05)

# Average travel time in 2022
m1 = df2.groupby(['hour'])['Trip Seconds'].mean().values / 30
s1 = df2.groupby(['hour'])['Trip Seconds'].std().values / 30
plt.plot(m1, color = 'red', linewidth = 1.8, label = '2022')
upper = m1 + s1
lower = m1 - s1
```

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

Tutorials

Intuitive understanding of core concepts in data science:

❶ Creating graphics

❷ Examining with realistic data

❸ Developing toy examples

❹ Providing Python implementation

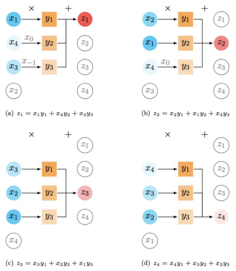
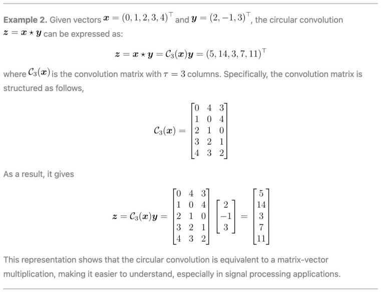

(a) $z_1 = x_1y_1 + x_4y_2 + x_3y_3$
(b) $z_2 = x_2y_1 + x_1y_2 + x_4y_3$
(c) $z_3 = x_3y_1 + x_2y_2 + x_1y_3$
(d) $z_4 = x_4y_1 + x_3y_2 + x_2y_3$
x_0
x_{-1}

Figure 2. Illustration of the circular convolution between $\boldsymbol{x} = (x_1, x_2, x_3, x_4)^\top$ and $\boldsymbol{y} = (y_1, y_2, y_3)^\top$. (a) Computing z_1 involves $x_0 = x_4$ and $x_{-1} = x_3$. (b) Computing z_2 involves $x_0 = x_4$. The figure inspired by Prince (2023).



Example 2. Given vectors $\boldsymbol{x} = (0, 1, 2, 3, 4)^\top$ and $\boldsymbol{y} = (2, -1, 3)^\top$, the circular convolution $\boldsymbol{z} = \boldsymbol{x} \star \boldsymbol{y}$ can be expressed as:

$$\boldsymbol{z} = \boldsymbol{x} \star \boldsymbol{y} = \mathcal{C}_3(\boldsymbol{x})\boldsymbol{y} = (5, 14, 3, 7, 11)^\top$$

where $\mathcal{C}_3(\boldsymbol{x})$ is the convolution matrix with $\tau = 3$ columns. Specifically, the convolution matrix is structured as follows,

$$\mathcal{C}_3(\boldsymbol{x}) = \begin{bmatrix} 0 & 4 & 3 \\ 1 & 0 & 4 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix}$$

As a result, it gives

$$\boldsymbol{z} = \mathcal{C}_3(\boldsymbol{x})\boldsymbol{y} = \begin{bmatrix} 0 & 4 & 3 \\ 1 & 0 & 4 \\ 2 & 1 & 0 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 14 \\ 3 \\ 7 \\ 11 \end{bmatrix}$$

This representation shows that the circular convolution is equivalent to a matrix-vector multiplication, making it easier to understand, especially in signal processing applications.

Source: https://spatiotemporal-data.github.io/posts/ts_conv

Reproducible Classes

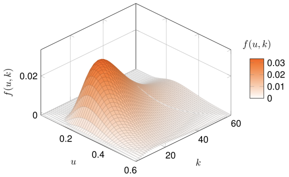
Open-source repositories:

❶ Providing supplementary materials

❷ Examining examples and codes

❸ Reproducing graphics in LaTeX

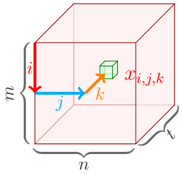
Visualization tool



`awesome-latex-drawing`

(1,300+ GitHub stars)

Explanation style



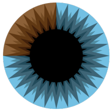
`awesome-beamer`

(100+ GitHub stars)

Class website will be available at `https://spatiotemporal-data.github.io`

References

- 3Blue1Brown
  - A math YouTube channel created and run by Grant Sanderson
  - Website: https://www.3blue1brown.com/
- Prof. Steve Brunton (UW)
  - Data-driven dynamics and control
  - Website: https://www.youtube.com/@Eigensteve


Source: Wiki

Thanks for your attention!

Any Questions?

About me:

- Homepage: https://xinychen.github.io
- MIT sites: https://sites.mit.edu/xinychen
- How to reach me: xinychen@mit.edu