Traffic Prediction in a Bike-Sharing System

Team 1
Jiaqi Liu, Shaowei Gong, Qiuyi Hong, Zhenhua Li, Caidan Liu
Background

- Bike-sharing systems are widely deployed in many major cities, providing a convenient transportation mode for citizens’ commutes.
- As the rents/returns of bikes at different stations in different periods are unbalanced, the bikes in a system need to be rebalanced frequently.
Solution Overview

Real-time monitoring

- Monitoring the current number of bikes at each station cannot tackle the challenge thoroughly, as it is too late to reallocate bikes after an imbalance has occurred.

Hierarchical Model

- Bipartite Station Clustering
- multi-similarity-based inference model
- check-in inference algorithm

Challenges:

- **Meteorology**
  - sunny/rainy
  - cool/warm
  - rare conditions

- **Correlation between stations**
  - nearby stations
  - mutual impact

- **Others**
  - time of day
  - day of the week
  - events,
Framework
Bipartite Station Clustering

- Group individual station into clusters according to their geographical location and transition patterns.
  - a single station’s traffic seems too chaotic to predict.
  - It is not necessary to predict the check-out/in of each individual station.
Bipartite Station Clustering

A) Iterative procedure of the bipartite clustering

B) An example of transit-matrix generation for a station $S_x$

$$\frac{N_jK_1}{n}$$

Worcester Polytechnic Institute
Entire Traffic Learning

- In hierarchical prediction model, the traffic in the higher level is predicted first.

- Time features
  - the hour of the day
  - the day of the week

- Meteorology features
  - weather
  - temperature
  - wind speed
Cluster Check-out Proportion Learning

- **Insights**
  - To allocate entire traffic to each cluster, we predict each cluster’s check-out proportion first
  - A multi-similarity-based inference model is proposed.

- Handle unbalanced meteorology distribution problem
- Guarantee cluster sum of 1 and manage between-cluster difference
Cluster Check-out Proportion Learning

- **Insights**
  - Our multi-similarity-based inference model integrates 3 similarity functions between features
  - Time similarity $\lambda_1 (t_1, t_2)$
  - Weather similarity $\lambda_1 (w_1, w_2)$
  - Temperature & wind Speed similarity $K ((P_{t1}, V_{t1}), (P_{t2}, V_{t2}))$
Cluster Check-out Proportion Learning

- Methodology
  - Assume 1, 2, ..., H are the H most recent periods to t
  - Denote corresponding check-out proportions $P_1, P_2, \ldots, P_H$ and $P_t$
  - Their features are $f_1, f_2, \ldots, f_H$
  - So that $P_t$ can be predicted by multi-similarity-based inference model

\[
\hat{P}_t = \frac{\sum_{i=1}^{H} W(f_i, f_t) \times P_i}{\sum_{i=1}^{H} W(f_i, f_t)}
\]
Cluster Check-out Proportion Learning

- Methodology
  - The multi-similarity function, $W(f_i, f_H)$ is obtained by

\[
\min_W \sum_{t=H+1}^T L(E_t \times P_t, E_t \times \hat{P}_t)
\]

  - $T \sim$ Sample size of historical data
  - $E_t \times P_t, E_t \times (\text{cap})P_t \sim$ ground truth and prediction value of check-out across clusters
  - $L \sim$ Loss function used to measure the prediction error
  - The multi-similarity function $W$ has 3 components:

\[
W(f_i, f_t) = \lambda_1(i, t) \times \lambda_2(w_i, w_t) \times K((p_i, v_i), (p_t, v_t))
\]
Cluster Check-out Proportion Learning

● Methodology
  ○ Time Similarity
    ■ Intuitively, check-out proportions corresponding to the same hour or a day are more similar than those corresponding to different hours
    ■ Additionally, if two proportion vectors both belong to weekdays or another, the more closed the two days are, the more similar these two vectors should be

\[
\lambda_1(t_1, t_2) = 1_{t_1, t_2} \times \rho_1^{\Delta h(t_1, t_2)} \times \rho_2^{\Delta d(t_1, t_2)}
\]
\[
\Delta h(t_1, t_2) = \min\{r(t_1, t_2), 24 - r(t_1, t_2)\}
\]
\[
r(t_1, t_2) = \text{mod}(|t_1 - t_2|, 24)
\]
\[
\Delta d(t_1, t_2) = \left[\frac{|t_1 - t_2|}{24}\right]
\]
Cluster Check-out Proportion Learning

- Methodology
  - Weather Similarity
    - The weather patterns are categorized into four categories: snowy, rainy, foggy and sunny
    - The similarity matrix is symmetric with 6 parameters

<table>
<thead>
<tr>
<th></th>
<th>snowy</th>
<th>rainy</th>
<th>foggy</th>
<th>sunny</th>
</tr>
</thead>
<tbody>
<tr>
<td>snowy</td>
<td>1</td>
<td>(\alpha_1)</td>
<td>(\alpha_2)</td>
<td>(\alpha_3)</td>
</tr>
<tr>
<td>rainy</td>
<td>1</td>
<td>1</td>
<td>(\alpha_4)</td>
<td>(\alpha_5)</td>
</tr>
<tr>
<td>foggy</td>
<td></td>
<td>(\alpha_6)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>sunny</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

\(\alpha_1 > \alpha_2 > \alpha_3, \alpha_4 > \alpha_5\)
\(\alpha_6 > \alpha_5 > \alpha_3, \alpha_4 > \alpha_2\)

- The more different two weather patterns are, the smaller the similarity between them is
Cluster Check-out Proportion Learning

- Methodology
  - Temperature/wind speed domain is continuous, with ‘missing’ scenarios in historical data
  - 2-D Gaussian Kernel function to measure the similarity between \((p_{t1}, v_{t1})\) and \((p_{t2}, v_{t2})\)

\[
K\left( (p_{t1}, v_{t1}), (p_{t2}, v_{t2}) \right) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{(p_{t1}-p_{t2})^2}{2\sigma_1^2} - \frac{(v_{t1}-v_{t2})^2}{2\sigma_2^2}}
\]

- As the prediction errors of successive time periods are not independent, we add an error correction item to the multi-similarity-based inference model.
Cluster Check-out Proportion Learning

● Methodology
  ○ The multi-similarity-based model adopted is

\[
\hat{P}_t = \frac{\sum_{i=1}^{H} W(f_i, f_t) \times P_i}{\sum_{i=1}^{H} W(f_i, f_t)} + \sum_{j=1}^{J} \psi_j e_{t-j}
\]

  ○ Here, the added items \( e_{t-j} = P_{t-j} - (\text{cap})P_{t-j} \) are the prediction errors of periods \( t - j, j = 1, 2, \ldots, J \); \( J \) is a threshold of time lag
Inter-cluster Transition Learning

- We predict each cluster’s check-in based on their check-out
Inter-cluster Transition Learning

- The inter-cluster transition matrix describes the transition probability between clusters.
- Using multi-similarity-based inference model to predict the matrix.
Trip Duration Learning

• In bike traffic, jam is no longer an important factor that affects trip duration

• It is mainly determined by the locations of bike stations

• Duration does not change too much
Trip Duration Learning

• According to NYC’s bike data, the trip duration between each pair of cluster

• By maximum likelihood estimation, we obtain symmetric matrix, describing the trip duration between cluster Ci and Cj
Online Prediction Process

• Check-out Inference
  – Entire traffic Prediction \( E_t \)
  – Check-out proportion prediction \( P_t \)

• Calculation
  – Check-out of each cluster \( C_i \) is
    \[ O = E_t \times P_t \]
Online Prediction Process

• Check-in Inference For Common Scenarios
  – use the same model as calculating check-out
    ▪ Entire traffic Prediction $E_t$
    ▪ Check-in proportion prediction $P_t$

• Check-in Inference For anomalous Scenarios
  – Update the prediction of target cluster in real time
    For a bike,
    o Original Cluster $C_i$
    o Check out time
    o Inter-cluster transition matrix and trip duration
  ▪ Get the expectation number of bikes on their way which are going to check in this cluster
Experiments

Data Source:

New York

We use the data of Citi Bike system, which is in NYC, from 1st Apr. to 30th Sep. in 2014 as the bike data. We use the meteorology data of NYC, from 1st, Apr. to 30th, Sep.

D.C

We use the data of Capital Bikeshare system, which is mainly in D.C., from 1st Apr. to 30th Sep. in 2014 as the bike data. We use the meteorology data in D.C., from 1st, Apr. to 30th, Sep., 2014
Baseline & Metric

Methodologies:
HA, ARMA, GBRT, HP-KNN, GC,

Metric:
RMLSE, ER
Results

Result of clustering

NYC

D.C.

GC

BC

GC

BC
## Results (cont.)

### Table 3. Prediction error of check-out across clusters

<table>
<thead>
<tr>
<th>Method</th>
<th>RMLSE</th>
<th>ER</th>
<th></th>
<th>Anomalous Hours</th>
<th>RMLSE</th>
<th>ER</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NY</td>
<td>WA</td>
<td></td>
<td>NY</td>
<td>WA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GC</td>
<td>BC</td>
<td></td>
<td>GC</td>
<td>BC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HA</td>
<td>0.387</td>
<td>0.372</td>
<td>0.439</td>
<td>0.451</td>
<td>0.353</td>
<td>0.355</td>
<td>0.453</td>
</tr>
<tr>
<td>ARMA</td>
<td>0.371</td>
<td>0.354</td>
<td>0.413</td>
<td>0.421</td>
<td>0.346</td>
<td>0.346</td>
<td>0.416</td>
</tr>
<tr>
<td>GBRT</td>
<td>0.386</td>
<td>0.369</td>
<td>0.423</td>
<td>0.425</td>
<td>0.311</td>
<td>0.314</td>
<td>0.371</td>
</tr>
<tr>
<td>HP-KNN</td>
<td>0.377</td>
<td>0.358</td>
<td>0.424</td>
<td>0.410</td>
<td>0.298</td>
<td>0.299</td>
<td>0.364</td>
</tr>
<tr>
<td><strong>HP-MSI</strong></td>
<td><strong>0.371</strong></td>
<td><strong>0.349</strong></td>
<td><strong>0.421</strong></td>
<td><strong>0.407</strong></td>
<td><strong>0.288</strong></td>
<td><strong>0.282</strong></td>
<td><strong>0.351</strong></td>
</tr>
</tbody>
</table>

### Table 4. Prediction error of check-in across clusters

<table>
<thead>
<tr>
<th>Method</th>
<th>RMLSE</th>
<th>ER</th>
<th></th>
<th>Anomalous Hours</th>
<th>RMLSE</th>
<th>ER</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NY</td>
<td>WA</td>
<td></td>
<td>NY</td>
<td>WA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GC</td>
<td>BC</td>
<td></td>
<td>GC</td>
<td>BC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HA</td>
<td>0.377</td>
<td>0.365</td>
<td>0.435</td>
<td>0.448</td>
<td>0.347</td>
<td>0.352</td>
<td>0.448</td>
</tr>
<tr>
<td>ARMA</td>
<td>0.363</td>
<td>0.352</td>
<td>0.409</td>
<td>0.418</td>
<td>0.340</td>
<td>0.344</td>
<td>0.405</td>
</tr>
<tr>
<td>GBRT</td>
<td>0.382</td>
<td>0.365</td>
<td>0.420</td>
<td>0.422</td>
<td>0.309</td>
<td>0.309</td>
<td>0.370</td>
</tr>
<tr>
<td>HP-KNN</td>
<td>0.375</td>
<td>0.360</td>
<td>0.415</td>
<td>0.411</td>
<td>0.302</td>
<td>0.295</td>
<td>0.367</td>
</tr>
<tr>
<td><strong>HP-MSI</strong></td>
<td><strong>0.365</strong></td>
<td><strong>0.350</strong></td>
<td><strong>0.408</strong></td>
<td><strong>0.402</strong></td>
<td><strong>0.297</strong></td>
<td><strong>0.290</strong></td>
<td><strong>0.353</strong></td>
</tr>
<tr>
<td>P-TD</td>
<td>0.384</td>
<td>0.373</td>
<td>0.425</td>
<td>0.419</td>
<td>0.335</td>
<td>0.302</td>
<td>0.365</td>
</tr>
</tbody>
</table>
Conclusion

Our model is better and applicable to different bike-sharing systems
Thank you!