HTTP: A New Framework for Bus Travel Time Prediction Based on Historical Trajectories

Wang-Chien Lee¹, Weiping Si¹, Ling-Jyh Chen², Meng Chang Chen²

¹ The Pennsylvania State University, PA, USA
² Academia Sinica, Taiwan
Observation

• Public transportation has become increasingly important.

• Customer Satisfactory is a high priority to various public transportation services.

• Many intelligent transportation services have been deployed in major cities world-wide, e.g., real-time bus information system.
Taipei e-Bus System

- Deployed since 2004
- 4,000+ buses, 300+ routes, 15 agencies
- Each bus sends its GPS location to information center every minute
- MANY applications…
Taipei e-Bus System (cont’d)
It looks perfect, but…

- MANY people complain the bus arrival time prediction is inaccurate.
- Government says sorry many times, but has no good solutions yet :<
- The prediction problem is very challenging, because there are many affecting factors, (e.g., traffic lights, congestion, accidents, and MRT stations) and the prediction error is accumulated for future bus stops.
Our Contribution

• We perform a detailed analysis to investigate the correlation between bus travel times in route segments and their temporal features based on historical data.

• We propose a new system framework, called HTTP, for travel time prediction of an on-going bus journey.

• We evaluate the proposed approach, in terms of prediction accuracy, using real bus dataset.
Related Work

- **Travel time prediction** [11, 22, 23, 34]
  - Mostly AI-based approaches that are computational expensive
  - Accurate for short-term prediction, but not for long-trips

- **Trajectory similarity** [2, 4, 7, 9, 12, 25, 26, 30]
  - Exploit conventional similarity measures for time series
  - Highly sensitive to errors and outliers

- **Trajectory patterns** [5, 13, 28, 33]
  - Classified trajectories in two aspects: trends and periodicity
  - Used to predict traffic condition of the same segment
Problem Formulation

• Bus route:

\[ R = \langle P_0, P_1, \ldots, P_n \rangle = \langle S_0, S_1, \ldots, S_{n-1} \rangle \]

where \( P_i \) is the \( i \)-th stop in a route, and \( S_i \) is the route segment between \( P_i \) and \( P_{i+1} \)

• Bus trajectory:

\[ T = \langle \langle P_1, t_1 \rangle, \langle P_2, t_2 \rangle, \ldots, \langle P_n, t_n \rangle \rangle \]

\[ = \langle \Delta t_1, \Delta t_2, \ldots, \Delta t_{n-1} \rangle \]

where \( \Delta t_i = t_{i+1} - t_i \) is the travel time on \( S_i \)

• Given \( T' = \langle \Delta t'_1, \Delta t'_2, \ldots, \Delta t'_i \rangle \) and \( R = \langle P_0, P_1, \ldots, P_n \rangle \), the travel time prediction problem is to predict \( \langle \Delta t'_{i+1}, \Delta t'_{i+2}, \ldots, \Delta t'_{n-1} \rangle \) in this trajectory
HTTP: System Overview

Bus Status Monitoring
Updated bus status and GPS data

Repository of Historical Trajectories

Speed, coordinates, timestamp

The HTTP Server
Travel Time Prediction
Prediction methods

Similar Trajectory Search
Patterns and indexes of historical trajectories

Query
Similar trajectory

Prediction parameters
Data Analysis

- **Dataset**
  - One 47.4km long bus route in Taipei
  - 24,985 trajectories from 2010/3 to 2011/3

- *The closer segments, the higher correlation!*
Data Analysis (cont’d)
Method 1: Passed Segments Scheme

- We extend the conventional notion of trajectory similarity by considering two trajectory similar when the difference for a segment is less than the threshold specified.

- To avoid exhaustive similarity comparison of all the trajectories in the repository, we partition the travel times for each segment into groups (using K-Means/V-Clustering).

- To efficiently identify similar trajectories using the travel times on passed segments of an on-going bus journey, we adopt the segment filtering approach to select similar trajectories.
Method 2: Temporal Features Scheme

• We consider two temporal features, “hours of a day” and “days of a week”, in historical dataset.

• We adopt the K-Modes algorithm to partition all the trajectories passing a segment into $K$ clusters using feature vectors and use the best matched cluster for prediction.
Method 3: Hybrid Scheme

• The PS and TF schemes are both based on historical trajectories, they can be easily combined into hybrid schemes.
  
  • Hybrid Passed segments/Temporal features (HPT)
    • Find a set of similar trajectories in terms of travel times of the passed segments and then filter using the temporal features
  
  • Hybrid Temporal features/Passed segments (HTP)
    • First apply TF and then PS
Evaluation

• Bus dataset from 2010/3 to 2011/2 as the historical dataset
• 508 trajectories in the first week of 2011/3 as the testing data
• Evaluate six schemes
  • Passed segments (PS)
  • Temporal features (TF)
  • Hybrid passed segments/temporal features (HPT)
  • Hybrid temporal features/passed segments (HTP)
  • TransDB [22]
  • Average prediction (AP)
Evaluation: Tuning the PS scheme

- PS w/ V-Clustering outperforms PS w/ K-Means.

- We suggest $V_{\text{thresh}} = 100K$, which is the maximum difference of weighted average variance allowed between two iterations.
Evaluation: Tuning the PS scheme (cont’d)

• **Window of segment filter (window)** is the number of the passed segments to look back, and we suggest window = 2.

• **Minimal number of trajectory (MNT)** is to ensure a minimum number of trajectories remained in the dataset, and we suggest MNT=200.
Evaluation: Tuning the TF scheme

- We suggest $K=3$ for the K-Modes scheme used in the TF scheme.
Evaluation: Comparison

The prediction error increases as the segment distance increases.

Normalized error: HPT/HTP < PS/TF < AVG/TransDB

HTP is slightly better than HPT -> Temporal Feature does matter!
Conclusion & Future Work

• Propose a prediction framework and develop two basic travel time prediction schemes

• Use a real dataset to evaluate the effectiveness of the proposed prediction schemes and produce good results

• Plan to explore more features and patterns from the historical trajectory data for the future work
Thank for your attention!