AN IMPROVED TAIPEI BUS ESTIMATION-TIME-OF-ARRIVAL (ETA) MODEL BASED ON INTEGRATED ANALYSIS ON HISTORICAL AND REAL-TIME BUS POSITION

Xue-Min Lu\textsuperscript{1,3}, Sendo Wang\textsuperscript{2}

\textsuperscript{1} Master Student, \textsuperscript{2} Associate Professor
Department of Geography, National Taiwan Normal University
\textsuperscript{3} Project Senior Technician
National Center for Research on Earthquake Engineering
TEL: +886-2-6630-0875, FAX:+886-2-6630-0858
E-Mail: xmlu@ncree.narl.org.tw

Abstract
In Taipei, current estimated time of arrival (ETA) is calculated only based on the distance and the estimated speed of the bus. In this research, the ETA model is improved by the historical and real-time bus position data. First of all, a spatial database of bus positions is established by receiving GNSS positioning data on each bus with a 25-second interval. The positions are corrected by map-matching algorithm. According to the difference of time and distance between each point, each period of bus travel time can be summarized as the basis of estimation. Using K Nearest Neighbor (K-NN) algorithm to search the most similar K records by the real-time bus position can served as the adjustment to the ETA model. Finally, the result shows that the proposed K-NN model can provide good accuracy, which the best mean absolute error is 140 seconds; and the best mean absolute percentage error is 7.44%.

Keywords: Estimated Time of Arrival(ETA), Global Navigation Satellite System(GNSS), Bus, Road Network, Real-time Traffic

INTRODUCTION

With the rapid development of Taipei city, the demand for public transport is obviously on the rise. In order to make the city’s transport network operate more efficiently, a reliable Intelligent Transportation System(ITS) is therefore necessary. Nowadays, nearly every bus in Taipei city has already been equipped with the Global Navigation Satellite System (GNSS) receiver and most of the bus stops have LED displayers installed accordingly to show the Estimated Time of Arrival (ETA). However, the calculation of the current ETA is solely based on the distance and the estimated speed of the bus, which is not sufficiently accurate and turns out to be a possible factor that leads to the unwillingness to take a bus. Thus, to solve the problem, this research is aiming at the historical and real-time bus position data as an alternative.

Before modeling the historical based ETA, some pre-processing is required to construct the historical database. First, we do map-matching algorithm to correct the GNSS data to match respective bus segments. M.A Qudus et al. (2007) discussed the limitations of a number of map-matching algorithms, including geometry, topology, probability theory, and other advanced map matching methods. Lou et al. (2009) took spatial and time factors into consideration, subsequently proposing the ST-algorithm to improve map-matching abilities. Strengthening on the ST algorithm, Hsueh et al. added the direction factor and formulated the STD-algorithm. The algorithms mentioned above are based on complex urban cases. In other words, They rely on the accuracy of matching a specific position in a complex urban spatial network. Differing from the research above, in our study, we duse routes data, which are provided by Taipei city Public Transportation Office, so that we can concentrate uniquely on temporal and directional factors. After building the historical database, we compare real-time data with historical data via K-nearest Neighbor (K-NN) algorithm. Liu et al. (2012) compared the K-NN with Artificial neural network (ANN) algorithm, finding out that K-NN produces better results. Chia-Ju Ho(2009) used the K-NN to test the K value, time window, and distance threshold setting with the buses in Taipei. He also found out that the square of the distance difference can make a better result. Yeh (2017) interpolated the historical data and re-adjust thresholds in different time period.

In this paper, we study buses in Taipei city, excluding the inter-city buses. Our GNSS position data are provided by Public Transport Data Exchange(PTX) which belongs to the Ministry of Transportation and Communications (MOTC).
The transmission delay of GNSS data’s return period is around 20 - 25 seconds. Finally, because of the timetable adjustment in April, the period of the route data are restricted between January 5th to the end of March.

**METHODOLOGY**

**map-matching algorithm**

The fields of GNSS data provided by PTX consist of plate number, route name, direction, azimuth, duty status, latitude, longitude, route unique identity and GNSS update time. Every location coordinate received from GNSS needs to match with a route line. Each route’s direction attribute includes outbound and inbound trajectories which are distinguished by the codes “0” and “1”. First of all, in order to simplify the operation, we convert each route into trajectory points whose interval is 3 meters. Secondly, we correct the GNSS points with the shortest distance, if a selected GNSS point has the same direction attribute with a route point. However, considering only the shortest distance between GNSS points and route trajectory points usually ends up producing fault matchings, especially when two buses pass the same stop in different directions. Therefore, taking azimuth attribute into account and introducing a weight in the calculation of distance may improve the outcome. As mentioned above, a weighted matching point is calculated with the following equation:

\[
\begin{align*}
  w_j &= \begin{cases} 
    0.25, & 0 \leq AD_{ij} < 30 \\
    0.5, & 30 \leq AD_{ij} < 60 \\
    0.75, & 60 \leq AD_{ij} < 90 \\
    1, & 90 \leq AD_{ij} < 180 
  \end{cases} \\
  d_{ij}^k &= \text{distance}(G_i^k, l_{ij}^k) \\
  P_i^k &= \min\{d_{i1}^k \times w_1, d_{i2}^k \times w_2, ..., d_{ij}^k \times w_j\}
\end{align*}
\]  

Here \( w_j \) is the weight of a route point \( j \) for the calculated distance; \( AD_{ij} \) is the difference between a GNSS point \( G_i^k \) and the azimuth of route points \( t \) to \( t+1 \). If the difference is bigger than 180, then \( AD_{ij} - 180 \); \( d_{ij}^k \) represents on a route named \( k \) the distance between a selected GNSS point \( G_i^k \) and a route point \( l_{ij}^k \); \( P_i^k \) represents the corrected point \( G_i^k \). Figure 1 shows an example of matching results. The rectangle in figure 1. is the nearest point between GNSS position and route point. Figure 2. shows the result we took azimuth into account.

![Figure 1. the result of map matching by only considering shortest distance.](attachment:image1.png)
Interpolate time and distance data

After running the map-matching algorithm, the next step is to build a database. In order to make time and position data searchable, it is necessary to interpolate data into every route point. First, we calculate the average velocity between every two consecutive GNSS positions. Then we use the time and the distance from the origin point to interpolate time into every route point. As a result, for any time range at a route point, we can estimate the moments when the buses pass by in the range. The interpolated results are shown in the table 1 below:

Table 1. interpolated data

<table>
<thead>
<tr>
<th>Plate Number</th>
<th>The distance from the origin point</th>
<th>11150</th>
<th>11195</th>
<th>11196</th>
<th>11199</th>
<th>11202</th>
<th>11205</th>
<th>11208</th>
<th>11211</th>
<th>11214</th>
</tr>
</thead>
<tbody>
<tr>
<td>269-U3</td>
<td>08:08:08</td>
<td>08:08:09</td>
<td>08:08:10</td>
<td>08:08:11</td>
<td>08:08:12</td>
<td>08:08:13</td>
<td>08:08:14</td>
<td>08:08:15</td>
<td>08:08:16</td>
<td></td>
</tr>
<tr>
<td>269-U3</td>
<td>21:00:44</td>
<td>21:00:47</td>
<td>21:00:49</td>
<td>21:00:51</td>
<td>21:00:54</td>
<td>21:00:56</td>
<td>21:00:58</td>
<td>21:01:00</td>
<td>21:01:03</td>
<td></td>
</tr>
<tr>
<td>269-U3</td>
<td>09:35:59</td>
<td>09:36:03</td>
<td>09:36:07</td>
<td>09:36:11</td>
<td>09:36:15</td>
<td>09:36:10</td>
<td>09:36:10</td>
<td>09:36:10</td>
<td>09:36:10</td>
<td></td>
</tr>
</tbody>
</table>

K-nearest Neighbor (K-NN) algorithm

The K-NN algorithm searches for the K records which share the most similar features from the historical database and then estimate the bus arrive time by analyzing these records. The idea is, primarily, to compare the input feature vector which we want to estimate with historical data in a time window of fixed length. The similarity is measured by distance metric. In our case study, we adopt the Euclidean distance. Finally, the K-NN algorithm averages the most similar K records to estimate bus arrive time.

The K-NN method is done as follows:

Step 1. Select a GNSS position $GP_m$ and retrieve the last n-1 records of GNSS positions before $GP_m$, which we denote by a vector $[GP_1, GP_2, ..., GP_n]$. n is a number which defines length of a vector.

Step 2. Calculate the time difference of the vector $[TD_1, TD_2, ..., TD_{n-1}]$. The result is denoted $[TD_1, TD_2, ..., TD_{n-1}]$. 

Figure 2. the result of map matching by taking shortest distance and azimuth into account.
Step 3. Find vectors from the same columns as \( GP_1, GP_2, \ldots, GP_n \) and calculate their time difference \( TD_{1T}, TD_{2T}, \ldots, TD_{(n-1)T} \) from the database if a time \( GP_{iT}^r \) is between \( GP_{nT}^r - T \) to \( GP_{nT}^r + T \). Here, \( T \) is the length of time window.

Step 4. Calculate the residual of each vector \( R^r = \sum_{w=1}^{n-1} (TD_{wT} - TD_{(n-1)T}) \) by distance metric, and pick the \( k \) smallest vectors. Figure 3 shows the Step 1 to 4:

![Figure 3. K-NN algorithm steps from 1 to 4](image)

Step 5. According to values of \( r \), get the travel time from the starting position to target position from the database, and average all the values to obtain the estimated travel time.

There are many parameters that can influence the estimations of the K-NN method. In this study, we discuss various \( K \) values and various lengths of time window \( T \).

**EXPERIMENT**

This research took buses in Taipei city as an example, except for highway buses. In order to verify the accuracy of this method, we have picked 11 bus routes. The routes that we picked are OSouth, 1, 20, 74, 218, 220, 270, 284, 588, 600, 611, and 949. Figure 4 shows the distribution of those routes. The Database features the GNSS data dated from January 5, 2018 to March 9, 2018. We verify the result with the real-time data from March 20, 2018 to March 31, 2018.

![Figure 4. Map of selected routes](image)
It is important for k-NN method to set parameters. In this experiment, The K value is tested for 5, 10, and 15; The length of time window is tested for 300, 600, and 900 seconds. Besides, this research also focuses on the difference between peak hours and off-peak periods. Hence, the periods between 8:00 to 9:00 and 17:00 to 18:00 are regarded as peak hours. On the contrary, the periods in 10:00 to 11:00 and 13:00 to 14:00 are regarded as off-peak ones.

The study used the Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) to evaluate the accuracy of our estimate. The formulas of MAE and MAPE are as follows:

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |e_i - o_i|
\]

\[
MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{e_i - o_i}{o_i} \right| \times 100\%
\]

Here, \(e_i\) is the forecasted value, \(o_i\) is the observed value, and \(N\) is the number of samples. MAE shows the average of the absolute differences between the forecast value and the observed value, MAPE shows the relative difference of the observed value and forecasted value.

**RESULT**

Our experiment tested a number of parameters for MPE and MAPE analysis. The K-value by 5, 10, and 15; the length of time window T tested by 300, 600, and 900 seconds. Starting and Ending points are randomly selected by computer. As the K-value increases, the number of Valid samples Less than or equal to origin ones. That is because some data are not long enough before a searching point. Table 2 shows the results of MAE and MAPE:

\[
\begin{array}{c|c|c|c}
\text{K} & \text{T} & \text{MAE} & \text{MAPE} \\
5 & 300 & 164s(N=339) & 8.46\% \\
 & 600 & 163s(N=321) & 8.39\% \\
 & 900 & 189s(N=299) & 9.25\% \\
10 & 300 & 140s(N=268) & 7.44\% \\
 & 600 & 141s(N=262) & 7.49\% \\
 & 900 & 158s(N=261) & 7.84\% \\
15 & 300 & 154s(N=237) & 7.57\% \\
 & 600 & 159s(N=231) & 7.79\% \\
 & 900 & 159s(N=229) & 7.80\% \\
\end{array}
\]

Where N is the sample sizes. According to table 3, the optimal estimation time can be found, when K equals 10 and T equals 600 seconds. Conversely when K equals 5 and T equals 900 seconds, the worst one was found. The difference between the best and the worst MAE is up to 49 seconds. In addiction, when the K value is 5, the T value has a greater variation; when the K value is 5 or 10, the T value has a less variation.

Finally, using parameters whose K equals 10 and T equals 600 seconds to test in difference of 4 periods. Table 3 shows the MAE and MAPE for the period from 8:00 to 9:00, 10:00 to 11:00, 13:00 to 14:00, and 17:00 to 18:00. The result shows that the MAE that we selected to represent the peak hour from 8:00 to 9:00 is up to 213 seconds, but the MAE from 13:00 to 14:00 is only 92 seconds. In other word, there is a significant difference between the peak hour and the off-peak period. This shows that this method should be improved if it is applied at different times.
Table 3. MAE and MAPE in peak hours and off-peak periods.

<table>
<thead>
<tr>
<th>Sample sizes</th>
<th>MAE</th>
<th>MAPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:00-9:00</td>
<td>24</td>
<td>213s</td>
</tr>
<tr>
<td>10:00-11:00</td>
<td>14</td>
<td>115s</td>
</tr>
<tr>
<td>13:00-14:00</td>
<td>34</td>
<td>92s</td>
</tr>
<tr>
<td>17:00-18:00</td>
<td>21</td>
<td>187s</td>
</tr>
</tbody>
</table>

In this case study, K-NN algorithm search similar historical data vectors in a range of time. It is effective and simple to estimate bus arrive time. And the estimated error can be about two minutes. Basing on GNSS position data as estimated resource, the accuracy of map-matchin algorithm affect estimation result. In order to get better one, except for imporoving the map-matching algorithm, there are more factors should be considered. For instance, driving on a bus lane or not, driving on holidays or weekdays, and so on.

CONCLUSION

This paper used K-NN algorithm to build a model for bus ETA. First of all, GNSS position data needed to be preprocessed by a map-matching method. The proposed method considered not only the shortest distance but also the azimuzh attribute to correct GNSS position to a bus route points. Secondly the time data are calculated and interpolated so as to make data searchable by each bus route points. Finally, the K-NN algorithm is applied to search historical database and the average of the most similar K records is taken as the ETA. By comparing K-value and time window T, in our case study, the result showed that when the K value is 10 and T is 600 seconds, the MAE attains the minimum 140 seconds and the MAPE is 7.44%. In addition, the study also revealed the differences between the results when the method is applied to the peak and the off-peak periods.

REFERENCES


