Traffic Velocity Prediction Using GPS Data: IEEE ICDM Contest Task 3 Report

Wei Shen, Yiannis Kamarianakis, Laura Wynter, Jingrui He, Qing He, Rick Lawrence, Grzegorz Swirszcz
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598
{wshen, yiannis, lwynter, jingruhe, qhe, ricklawr, swirszcz}@us.ibm.com

Abstract—This report summarizes the methodologies and techniques we developed and applied for tackling task 3 of the IEEE ICDM Contest on predicting traffic velocity based on GPS data. The major components of our solution include 1) A pre-processing procedure to map GPS data to the network, 2) A K-nearest neighbor approach for identifying the most similar training hours for every test hour, and 3) A heuristic evaluation framework for optimizing parameters and avoiding over-fitting. Our solution finished Second in the final evaluation.

Keywords-map-matching, nearest neighbor, cross validation

I. TASK DESCRIPTION

As the planet becomes more instrumented, cities across the world are able to collect transportation data from numerous sources in real-time. By combining traffic data collected from different sources with advanced algorithms for traffic prediction, both traffic management centers and individual drivers can make smarter choices and real-time traffic management and trip planning. The IEEE ICDM Contest 2010, sponsored by TomTom, addresses a critical issue in transportation systems: How can we predict traffic congestion by making use of available real-time data?

The contest consists of three prediction tasks, each of which aims at predicting a different traffic congestion measure (i.e., traffic flow, traffic jams, and link travel speeds) for the city of Warsaw, Poland, based on traffic data generated from a vehicular traffic simulator. Task 3 is particularly focused on the data that comes from GPS devices and combining that with another source of more aggregate traffic data providing typical link speeds by time of day and day of the week. In particular, Task 3 involves traffic reconstruction and prediction based on simulated real-time information collected from the simulated GPS devices of individual drivers. The training data of task 3 consists of two files: one contains the stream of notifications from 1% of vehicles in the network about their current GPS locations in the city road network, sent every 10 seconds, and the other provides the actual average velocities in corresponding simulation cycles and time periods, on 100 selected road segments. Fifty 10-hour long simulation cycles are covered in the training data set, whereas test data cover another 500 hours of simulation, split into 1-hour windows. The challenge is that the stream of GPS notifications for the first half of each 1-hour window in the test data is revealed, but not the second half. Participants are asked to predict the harmonic-average speed of vehicles that will be passing the 100 segments in two time periods: 0 - 6’ and 24 - 30’ of the second half of each such 1-hour window.

The following four features make task 3 particularly challenging:

- The size of the data set is huge (several GB uncompressed). An efficient method is needed to map the data set of GPS coordinates to links of the network.
- The sampling rate of vehicles varies from link to link and is in general very low on the 100 selected road segments.
- Simulation hours during the “warming-up” period of a simulation cycle exhibit different properties from hours in the middle of a cycle. However, it is not known in the test data which hours were extracted from a “warming-up” period versus from the rest of the cycle.
- The simulated data set has some distinct features which make temporal-spatial regression models for traffic prediction fail in this specific context.

Solutions of Task 3 are evaluated by calculating the Root Mean Squared Error (RMSE) of the inverse of the predictions so as to represent the RMSE of travel times, rather than of speeds. Predicted speeds are hence transformed to - through inverting and multiplying by 60 - into predicted travel time over 1 km of the road segment, expressed in minutes. These travel times are compared with the simulated ground truth using RMSE. Our final solution has a RMSE of 7.4556 on the test data set, which resulted in second place in the contest. The remainder of this report discusses the major steps we took to achieve this solution.

II. MATCHING GPS DATA TO THE NETWORK

Each GPS record in both the training and test data sets contains time, an id, latitude/longitude coordinates, and velocity of the probe vehicle, but most importantly is not associated with the links of the network. Before any data mining methods can be performed, a pre-processing procedure is needed to map the GPS records to the map. GPS data in reality are usually noisy and the reported coordinates may not fall exactly on any links of the network. The procedure of deciding which link of the network that a GPS point is located on is called map-matching in the Geographic Information System (GIS) community.

Over the years, researchers have developed diverse map matching algorithms (e.g., Greenfeld 2002, Alt et al. 2003,
etc.). In our context, GPS signals come from simulated traffic data and are therefore less noisy than real GPS signals. To perform the map-matching, we used the built-in overlay function of the software package ArcGIS. First, coordinates of the Warsaw map were loaded into ArcGIS, resulting in a link layer of the map. Then, GPS data of both the training and test sets were imported and joined with the link layer by searching the closest link for each point (Figure 1). Note that for bidirectional links such a method may map a GPS point to the wrong direction. Fortunately, our investigation revealed that bidirectional links in the network are mostly minor links, whose impact to the overall traffic pattern in the network is less significant as compared to the major links. We believe this potential deficiency of the method did not have a critical impact on the final prediction accuracy.

The map-matching procedure was performed for every simulation cycle in both the training and test data sets. After map-matching, each GPS data point then had a link ID associated with it. The new data set was then loaded to a PostGreSQL database for further processing.

III. THE K NEAREST NEIGHBOR MODEL

An effective approach for real-life road traffic prediction is through specialized auto-regressive models, in which measurements of the traffic on the link of interest as well as on certain neighboring links are used as input (see, e.g., Min et al. 2007). Unfortunately, that method does not work well for this simulated data set, due in a large part to the following reasons:

- The relatively low GPS sampling rate (1% overall) makes it impossible to construct reliable historical speed profiles for all the links. For example, for each of the selected 100 links of interest, the total number of GPS points during the first half an hour vary from 0 to 934, with mean = 17 and standard deviation = 53.2. In fact, 22 links among the 100 selected road segments have no GPS data points at all.
- The actual average link velocity provided by file 2 of the training data shows that the simulated speed profile on the 100 links typically involves sudden drops or sudden rises (Figure 2).

Instead, we devised a completely different approach to predict the traffic speeds from the GPS points. The approach works by constructing a K-nearest neighbor model to predict vehicle velocity. Namely, for each test hour, we pick the training hours that are most similar to it and use a linear combination of the corresponding velocities for the first and last 6-min intervals during the second half of the hours as its estimate.

The following two criteria are used to construct the similarity measure.

1) Global similarity \( S_{i,j}^g \); \( S_{i,j}^g \) measures how close the total number of GPS counts in one test hour is to that of a training hour. The total number of GPS points in the network reflects the overall congestion level of the entire network, and hence is a good indicator of whether an hour is during the “warming-up” period of a cycle or not. To construct \( S_{i,j}^g \), let \( c_t^i \) and \( C_t^j \) be the total number of GPS points received during every 1-min interval \( t = 1, \ldots, 30 \) of test hour \( i = 1, \ldots, 500 \) and training hour \( j = 1, \ldots, 500 \), respectively. The global similarity between a test hour \( i \) and a training hour \( j \), denoted as \( S_{i,j}^g \), is measured by the RMSE of \( c_t^i \) and \( C_t^j \). Namely,

\[
S_{i,j}^g = \sqrt{\frac{\sum_{t=1}^{30} (c_t^i - C_t^j)^2}{30}}
\]  

2) Local similarity \( S_{i,j,k}^{11} \) and \( S_{i,j,k}^{12} \): Comparing the GPS records with the actual harmonic average speed provided in the training data, we found that during a 6-min interval on a selected road segment, the speed of one probe vehicle can be significantly different from another. For instance, one GPS record may show a speed instance of zero while another one may report speed = 60km/hr. The huge variance in sample speed is partly due to the discrete feature of the traffic simulator. As a result, the harmonic average velocity of probe vehicles does not generally lead to reliable velocity estimates. In fact, it is often impossible to take the harmonic mean of the speed of the probe vehicles, as many probe vehicles report speed values of zero in the GPS data.
Nevertheless, the average link speed and GPS data on the link do exhibit a strong correlation as follows: typically, links with low speed have many more GPS records with zero values, whereas links with high speeds are more likely to have nonzero GPS records. This observation motivated us to construct a local similarity measure based on the total number of GPS records with zero and nonzero values on any link \( k \) of interest. Interestingly, we did not get improvement from looking more closely at the values of the non-zero speeds, as one might expect. Hence, the local similarity \( S_{ij}^{l1} \) and \( S_{ij}^{l2} \) measuring the similarity of a test hour \( i, i = 1, \ldots, 500 \) and a training hour \( j, j = 1, \ldots, 500 \) on link \( k, k = 1, \ldots, 100 \), is computed as follows:

\[
S_{ij}^{l1} = |p_i - P_j|, \quad S_{ij}^{l2} = |q_i - Q_j|
\]

where \( p_i \) and \( P_j \) are the total number of GPS records with zero values during the first half of test hour \( i \) and training hour \( j \), respectively; \( q_i \) and \( Q_j \) are the total number of GPS records with nonzero speeds. For each candidate hour \( j \), the actual similarity measure \( S_{ijk} \) for each training hour \( i, i = 1, \ldots, 500 \) is computed as the weighted sum of the ranks of the global similarity and the local similarities. Namely, \( S_{ijk} = \alpha_k \text{rank}(S_{ij}^{g}) + \beta_k \text{rank}(S_{ij}^{l1} + \gamma_k \text{rank}(S_{ij}^{l2}) \)

Note that the rank of a training hour is measured by its position when the corresponding similarity measure for all training hours is sorted in ascending order.

Finally, the harmonic average speeds of the first and last 6-min intervals of the second half of each test hour are estimated as the weighted harmonic average speeds of the corresponding intervals of the \( K \) most similar training hours. The inverse of the similarity metric of each candidate training hour is used as the weight. In fact, three potential estimators, the arithmetic mean, the median, and the harmonic mean, were tested when we construct our solution. Overall, the harmonic mean provided the best result.

One potential problem of using the harmonic mean of the \( K \) nearest neighbors is that if all the candidate hours in the neighbor list have high speed except for a few small outliers, the harmonic mean can be very small. The existence of such cases contributes to quite a significant portion of the error. To avoid the outlier effect, a conditional trimmed harmonic mean is used by filtering out the rare small outliers when most of the neighbors have high velocity values.

IV. THE EVALUATION FRAMEWORK

For each link \( k \) of interest, our \( K \) nearest neighbor method with the outlier filter has seven parameters in total:

1) \( K \) - the total number of neighbors used in constructing the velocity estimate;
2) \( \alpha_k \) - weight of the global similarity measure;
3) \( \beta_k \) - weight of the local congestion similarity measure;
4) \( \gamma_k \) - weight of the local free flow similarity measure;
5) \( n_k \) - the total number of high speed neighbors for the outlier filter to be initiated;
6) \( h_k \) - the high cut-off value of the outlier filter;
7) \( l_k \) - the low cut-off value of the outlier filter.

All of the above parameters are optimized heuristically using a 5-fold cross validation framework. A set of parameters were regarded as optimal if it generated the best average performance over the five test-training data sets. Finally, the link-specific optimal parameter settings were applied to the real test data to obtain the final solution for task 3. We found that it is often the case that the actual performance measure (7.4556) on the real test data set is slightly better than the average best performance measure (7.74) from the cross validation. This is understandable as cross validation only uses 4/5 of the training data.

V. CONCLUSIONS

The simulated link speed profile involves more dramatic changes than those typically observed in reality, and hence is much more difficult to predict. Overall, our \( K \) nearest neighbor model combined with the outlier filter is robust, scalable, and generates satisfactory results (ranked No. 2 in the final evaluation). The following key observations lead to the effectiveness of our model:

1) The total number of GPS records in the network provided a good indicator of the position of a simulation hour during simulation cycles;
2) Constructing the velocity estimate based on file 2 of the training data instead of the GPS stream in file 1 as the latter is very noisy and scattered;
3) Using the total number of GPS counts with zero velocity on each link as a major component of local similarity;
4) Taking the harmonic mean of the nearest neighbors instead of the arithmetic mean or median as the estimate.

Due to time constraints, we were unable to extend the method to incorporate information from other links in geographic proximity to the 100 test links, an extension which may serve to increase the quality of the predictions further.

ACKNOWLEDGMENT

We thank the contest organizers and sponsor of the contest for the interesting and challenging competition.

REFERENCES

