Taxi Trip Time Prediction Using Similar Trips and Road Network Data

Aakash Deep Singh  
Department of Mathematics and Computing  
Indian Institute of Technology, Delhi  
mt5110581@maths.iitd.ac.in

Wei Wu, Shili Xiang, Shonali Krishnaswamy  
Data Analytics Department  
Institute for Infocomm Research  
wwu, sxiang, spkrishna@i2r.a-star.edu.sg

I. INTRODUCTION

Taxi travel time prediction is very important for services providers and the end users. Lots of solution have been proposed for estimating travel time using the historic GPS trajectories [1], [2]. But not everyone has access, resources or means to store this huge trajectory data. In the absence of these GPS trajectories the problem for predicting travel time becomes difficult.

The travelling behaviour of humans follows a certain pattern. Many people travel between similar locations at similar time of day and day of week. This results in almost similar traffic conditions which in turn results in similar travelling speeds and trip times. We exploit this fact to build a system to predict travel time of a taxi trip.

We also observed that most of the trips are made up of three major parts: an initial part to reach a highway from the starting location, then moving toward the destination along this highway, and then from the highway to the destination. To model the time taken by a trip to complete, it is sufficient to model the time across these three components of a trip. The time taken to move across the highway is almost same to other taxis that moves in similar traffic conditions.

Our system comprises of two major components: similar trips mining and frequent sub-route mining. The first component finds the trips that are geographically and temporally similar to the trip for prediction. Instead of the normal approach of finding k-nearest trip for every query, we propose to find spatio-temporal clusters of trips in the history. The frequent sub-route mining component uses road network data from OpenStreetMap(OSM) to find out shortest path trajectories and use them to discover frequent sub-routes.

II. SYSTEM

A. Framework

Figure 1 shows the framework of our system. The left hand side is the offline part which comprises of training the model, whereas the right hand side part is online part which will handle user queries (e.g., trips for travel time prediction). The training part itself is divided into two parts, one is finding similar trips while the other is sub-trajectory clustering.

B. Finding Similar Trips

For finding trips that are similar in nature, one needs to consider whether two trips are similar spatially, i.e., which starts and end in close by regions, and faced similar traffic conditions on the way. To find trips which are spatially close we did a 4-D clustering using the origin and destination coordinates of the trips. To model the traffic with this data, we further divided each spatial cluster into temporal clusters. To do this, we divided each days of the week in 48 periods of 30 minutes each. We clustered these time periods according to the mean time taken by the trips across each periods.

To achieve this clustering on the huge trip data at our hand, we needed an algorithm that is both scalable with number of samples and number of clusters. Moreover, we needed an algorithm that can be used for predicting also as we would like to know the clusters for the query trips. On the top of it, the clustering algorithm should not be com-
Figure 2: Showing time labels of a spatial cluster after clustering over means. It is evident that the time taken in the period 25 to 40 is different than the rest of the day indicating different traffic conditions during the time.

Computationally expensive. So, we used MiniBatchKMeans[3] algorithm for the process as we could work with small batch size which could fit in small system memory and the overall trained model can also be used for prediction. From this point onwards, we analysed each spatial cluster individually.

To have a measure of traffic, the average time for trips were taken across the periods of the days. If the traffic is similar among two different periods of a particular spatial cluster, the average travel times of both the periods will be close. So, to accommodate this, we re-clustered these spatial clusters obtained in the previous steps using the average time of trips for the period as a feature. Figure 2 shows an example of 1 such clustering. We trained a regularised classification model over these temporal clusters to avoid the problems of data sparsity as there might not be trips in every period of the week across some cluster. It also prevents overfitting by stopping few outlier trips to decide the traffic conditions. For a new query trip, finding it’s neighbours is just finding in which cluster it belongs to, which can be done very efficiently, and fetching the trips from the history belonging to the same spatio-temporal cluster.

C. Sub-Trajectory Clustering

In this part, we focus on individual groups. We use the OSM data, a user-generated open source map, to find the shortest path route between all the trips which belongs to same spatio-temporal cluster. In the absence of original trajectories of these trips, the shortest path trajectory from OSM is a reasonable approximation of the actual path. For a single spatio-temporal cluster, we cluster all these points to get the sub-trajectories among this group. After this, we map each trip to these sub-trajectories and get three components for each: the “first-mile” that from source to highway, the frequent sub-trajectories on highways, and the “last-mile” from highway to destination. To predict a time for a trip, a regression model was trained over trips mapped as these three components.

We used the routing service by OSRM[4], an open source routing engine designed to run on OpenStreetMap data. OSRM gives us sequence of GPS points of the shortest path between origin and destination. It gives a recurrent set of GPS points for trips consisting a common path which means certain GPS points will appear multiple number of times in the shortest path trajectories generated by it. The points that are travelled together must be spatially close and are travelled almost equal number of times. Which means that the points over the intermediate highway will have almost similar frequency. We used this fact to find the common sub-trajectories. We used the latitude, longitude and frequency of the points to find the approximate clusters. As the scale for frequency is different from the other two features, clustering them straight away did not resulted any good clusters. Also, the points provided by OSRM routing engine for the roads were not at uniform distance. Mostly the points on highways are mapped at greater distances than the other parts.

To overcome this problem, we needed to divide these GPS points along with their respective frequencies. For subsequent discussion we will call this group OSM-points. These OSM-points were divided into groups according to their frequencies and further clustered using Density based clustering algorithm. Also, as the frequency of these groups increased, the eps parameter for the DBScan[5] algorithm was also increased. This took into account the non-uniformity of distances among the OSM-points as there is a direct correlation between the points on highway and their frequency. These DBScan parameters will vary according to the map data in hand. The value of this eps parameters were selected experimentally to fit the road network of Singapore. To make this approximate algorithm more robust, the sub-trajectory clusters were passed to a cleaning process which divided them further if the points among them were not occurring in a continuous sequence of points in a path generated by OSM. These sub-trajectory clusters were given unique numbers and stored in memory. Figure 3 shows few examples of such clusters.

D. Prediction Model

All the trips in a spatio-temporal clusters were mapped as a sequence of numbers using the labels of sub-trajectory
clusters which appeared in the OSM generated path. Distances were calculated from the source of the trip to the first sub-trajectory cluster and from last sub-trajectory cluster to the destination point using the OSRM routing engine. Now the trips were mapped to an initial distance, sequence of sub-trajectory labels, and distance from last label to destination. This data was passed to a Support Vector Regression (SVR) [6] system to learn the time taken by each component of the trip. This model was stored in the memory.

E. Handling Queries

For a test query, first a spatio-temporal cluster was predicted using already fitted models. Using the OSM, shortest path was calculated. The sub-trajectory clusters and the SVR model stored in the memory were loaded with respect to the spatio-temporal clusters. After mapping this new trip to a pair of distances from trip-dependent-path and a sequence of sub-trajectory labels, the travel time was predicted.

III. PERFORMANCE STUDY

We used a trip dataset containing more than 12 million trips across Singapore over a period of 30 days containing origin, starting time, destination, and ending time. We selected 100,000 random trips for testing and used the rest for building the model.

RMSE was used as an accuracy measure for the model. The number of clusters ($K$) is an important parameter of the system. The RMSE decreases with increasing $K$, but the changes in accuracy is small after $K = 14000$. As a result of this, we chose $K = 14000$ as the optimal number of clusters for the model.

We studied the impact of sub-trajectory clustering on accuracy of trip time prediction. We find sub-trajectory clustering help reduce the prediction error. Figure 4 shows the decrease in average error in various distance ranges.

Figure 5a presents the performance change of the model with time and day of the week. It can be observed that the error is almost similar for the first 4 working days of the week along with the first half of Friday, which is very intuitive. It also shows that the model is not able to capture the travelling times for Friday especially in the evening. A possible reason for this can be the unpredictable nature of human movement on the evening. Figure 5b and 5c, shows the variation of average error with days and periods of the day respectively.

IV. CONCLUSIONS

Trip time prediction is an important problem. Taxi passengers often want to know when they will arrive at their destinations. We design a method of predicting taxi trip time by finding historical similar trips. Trips are clustered based on origin, destination, and start time. Then similar trips are mapped to road networks to find frequent sub-trajectories that are used to model travel time of the various parts of the routes. Experimental results show this method is effective.

REFERENCES


