Road vehicle speed estimation using dense GPS trajectory data

Ragnar Vent
Institute of Computer Science
University of Tartu
rvent@ut.ee
Supervisor Amnir Hadachi

ABSTRACT
The high availability of GPS-enabled devices and the large amounts of data gathered from them gives us an opportunity to get a better insight into the state of the traffic and the road network. The present work was motivated by the sample of T-Drive GPS trajectory dataset made available by Microsoft Research in [1]. The ultimate goal was to use the given trajectory data for estimating the average speeds of the road sections and thereby get an overview of the road network in terms of speed.

1 INTRODUCTION
The abundance of GPS tracking data combined with digitized map data provide an excellent basis for road traffic analysis, which can be exploited for urban planning, traffic management and other purposes.

The goal of the present work was to use GPS trajectory data gathered for the T-Drive project for road traffic analysis. T-Drive itself is a driving direction services based on GPS trajectories of a large number of taxis and it helps the user to find out the practically fastest path to a destination at a given departure time[2]. The service was built based on real-world taxi trajectory data that was gathered over the span of 3 months. The dataset has since been used in scope of different works [3][4]. A sample of that dataset was made freely available and was used in the context of this work. The aim was to use the T-Drive sample dataset to perform a speed analysis of the Beijing road network.

As a first step we preprocess the trajectory data in order to remove outliers, inconsistencies and irrelevant data(see Section 2). Next we associate the trajectory points to roads sections using map-matching algorithm described in Section 3 and calculate the average speed for each road section. Finally we generate a map based on the used road network. In order to investigate the effect of time on the road section speeds, the analysis is done separately for each day and for different time periods during a day. In Section 5, we use an approach based on Kalman filter to get a speed estimate for the next point in trajectory data, given that we know the current speed, location and the location of the next point. Based on the results, we recalculate the the average road section speeds and compare them to the result we got with real values.

2 DATA GATHERING AND PREPROCESSING
The road network data was taken from OpenStreetMaps. The map itself comes in XML format and consists of nodes and ways. Each node is defined by its ID and its geodetic location(latitude and longitude). A way is comprised of nodes and can have multiple tags attached to it(e.g name, what type of road it is, whether it is one-way). In order to speed up XML parsing, the road data was first filtered with Osmosis[5]. All irrelevant information(e.g railways, footways) was removed during filtering. In addition the map was reduced to a smaller area in the central Beijing(bounded by latitude [39.8146, 40.0339] and longitude [116.1687, 116.5495]), since working with the entire map of Beijing is quite time consuming in terms of map-matching and road network parsing.

The T-Drive sample dataset used for speed estimation analysis consists of 10357 taxi trajectories with a total of over 15 million points. The data was gathered during the period of Feb. 2 to Feb. 8, 2008 in Beijing. A sample of the trajectory data is given in Fig. 1.
For the GPS trajectory data we remove duplicates and data that has low-sampling rates, unrealistic speeds or its coordinates are out of given bounds. The following filters were applied to the data.

- **Duplicate** A trajectory point was considered a duplicate if it had the same timestamp as the previous point.

- **Long sampling interval** If the time between two sequential points was more than 20 seconds then the trajectory segment was removed. The goal was to work with dense data, since with low-sampling rates we would have had to use some kind of shortest-path algorithm to determine the distance between the points for speed estimation.

- **High speed** Speed higher than 90 km/h between two points was considered unrealistic and in that case the trajectory segment was removed.

- **Coordinates out of bounds** All points that were outside of the bounds defined by the road network were removed.

3 MAP-MATCHING AND SPEED ESTIMATION

In order to associate trajectory points to road sections, a variation of point-to-curve map-matching algorithm described in [6] was implemented. The general idea of the algorithm is as follows:

- Perform a query through all the nodes in the road network and find the one that is the closest to the trajectory point. Since the number of trajectory points needed to be associated with roads ranged in millions, a kd-tree[7] data structure was used in order to speed up the search process.

- Find all the road segments that are incident to that node

- Calculate the distances to each of the road segments and choose the closest one.

The distance between two GPS points was calculated using Haversine formula:

\[
a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)
\]

\[
c = 2 \cdot \tan^2\left(\sqrt{a}, \sqrt{1-a}\right)
\]

\[
distance = R \cdot c
\]

Where \(R\) is radius of earth, \(\phi\) latitude and \(\lambda\) longitude. Haversine formula was mainly used during speed estimation part, in some other instances (e.g closest node query) a simple Euclidean distance was used instead. The distance from a trajectory point to a road segment, defined by two nodes in the road network, was calculated using cross-track distance formula:

\[
d = \sin((\delta_{13}) \cdot \sin(\theta_{13} - \theta_{12}) \cdot R)
\]

Where \(\delta_{13}\) is the distance from the road segment start point to the third point, \(\theta_{13}\) is the initial bearing (also known as forward azimuth) from start point to the third point and \(\theta_{12}\) is the initial bearing from road segment start point to its end point.

In case of speed estimation, two sequential trajectory points had to be associated to the same road section in order for the speed to be determined. All the speeds linked to the same road were then averaged. The process was repeated for all the trajectories and afterwards an overall average was calculated for each road section.

The road speed analysis was done using Java JDK 1.8(repository at [8]). After the average speeds were determined for the road sections, the results were then visualized using Java minigeo library.

4 SPEED ESTIMATION RESULTS

After preprocessing about 3.8 million trajectory points were left. The road network that was used in speed estimation consisted of 25391 nodes and 7876 road sections after filtering with Osmosis. The road speeds were split into the following three categories for visualization purposes:

- **Fluid** - \(speed > 30\frac{km}{h}\)
- **Dense** - \(speed \in (20, 30]\frac{km}{h}\)
- **Congested** - \(speed \leq 20\frac{km}{h}\)

Fig. 2 provides a spatial visualization for the average road speed estimation results.
The average speed was determined in total for 6402 road sections, of those 669 number of road sections were determined as fluid, 1171 as dense and 4562 as congested. As a next step, we investigate the road section speeds per modality. Since the data was gathered over a period of 7 days, we can focus on each day separately. Additionally we split the 24h timeline of the day in 6 parts(see Fig. 4) and perform the speed analysis for each part separately.

Figure 3: Speed distribution of central Beijing road network

The mean speed of the road sections in central Beijing is $17.313 \frac{km}{h}$ and has a standard deviation of $10.351 \frac{km}{h}$. The distribution of the estimated speed values can be seen in Fig. 3.

Figure 4: Different time intervals during a day for which the speed analysis was performed

The most fluid part of the day in terms of speed was during 06:00-09:00, for which the mean of road section speed was recorded highest for 5 different days(Day 7 06:00-09:00 had the highest mean of $29.869 \frac{km}{h}$). On the other hand 14:00-18:00 time interval had the lowest mean of road speeds(recorded lowest for 4 different days).
5 KALMAN FILTER

Kalman filter is a recursive data filtering algorithm, which incorporates all the information provided to it to get an overall best estimate, such that the Mean Square Error (MSE) is minimized. The goal is to use noisy measurements and (hopefully) produce more accurate estimates. Essentially, we use all the available measurements (e.g., location, velocity, acceleration) in the current state and the knowledge of the system to predict the measurements at the next state. Once we get the real measurements for the next state, we use those and the predicted values to get the best estimate. This makes Kalman filter especially useful for real-time data processing. The actual Kalman filter formulas and their derivations have been the subject of many works, such as [9][10] and have been used to an extent for present work.

The goal is to apply the Kalman filter on the taxi trajectory data and calculate Kalman estimates based on given two trajectory points. The general idea is that if we know the speed and location of the current state (i.e., some trajectory point) and the location of the next state, we can use the Kalman filter formula to get an estimate for the speed at the next state. In order to calculate a Kalman estimate based on two sequential trajectory points, the initial speed is first projected to velocities on different axes and the respective geodetic coordinates are converted to UTM coordinates in meters.

Fig. 5 shows the road network with regard to speed by first performing 100,000 Kalman estimations (overall MSE was $1436.6 (\frac{\text{km}}{h})^2$). The initial trajectory points based on which we calculated the Kalman estimates were chosen randomly from the entire filtered trajectory data. The overall visual changes appear to be minor, as such, some of the differences compared to the results with real estimates (see Fig. 5) have been highlighted with bounding boxes.

Figure 5: Difference in road section speed estimation with real estimates (left image) and with 100,000 Kalman estimations (right image)
However, in order to get a better sense of the Kalman estimates, MSE was calculated per road section basis. By looking at Fig. 6, we can see significant fluctuations, in some cases reaching over $4000\,(\text{km/h})^2$. Nonetheless, for most cases the MSE stayed below $1000\,(\text{km/h})^2$. In addition, the Kalman estimate turned out to be higher than the real estimate quite frequently, indicating that by increasing the random number of Kalman estimations, the overall image will appear much more fluid in terms of speed.

6 CONCLUSIONS

In this work, we perform a speed analysis of central Beijing road network based on real-world taxi trajectory data. We start by preprocessing the trajectory data and then associating the points to road sections using point-to-curve map-matching algorithm. During road section speed estimation we only use dense data (sampling interval $< 20\,s$). The resulting image reveals that bulk of the roads are determined as congested. This may be affected due to the fact that the current solution does not take into account traffic signals, which may especially come out in smaller midtown streets, where the traffic is highly regulated with signals. In addition we look at the results per modality basis by splitting the day in 6 parts and looking at each day separately. The results show that the traffic congestion is at its highest during 14:00-18:00 and lowest during 06:00-09:00. In the last part we calculate Kalman estimates for different trajectory points at random and compare them to the real estimates. The subsequent result shows that the overall image doesn’t change significantly.

7 REFERENCES

Figure 7: Overview of central Beijing road network with regard to speed at day 1
Figure 8: Overview of central Beijing road network with regard to speed at day 2
Figure 9: Overview of central Beijing road network with regard to speed at day 2 - continued

Figure 10: Overview of central Beijing road network with regard to speed at day 3
Figure 11: Overview of central Beijing road network with regard to speed at day 3 - continued
Figure 12: Overview of central Beijing road network with regard to speed at day 4
Figure 13: Overview of central Beijing road network with regard to speed at day 4 - continued

Figure 14: Overview of central Beijing road network with regard to speed at day 5
Figure 15: Overview of central Beijing road network with regard to speed at day 5 - continued
Figure 16: Overview of central Beijing road network with regard to speed at day 6
Figure 17: Overview of central Beijing road network with regard to speed at day 6 - continued

Figure 18: Overview of central Beijing road network with regard to speed at day 7
Figure 19: Overview of central Beijing road network with regard to speed at day 7 - continued