CS229 Project Report–Complex Overpass Reconstruction Based on Taxi GPS Data

Yilong Geng

1 Abstract

In this project we try to reconstruct detail-complex overpasses with sparsely sampled taxi GPS sequences. The goal is to learn about the shape and direction of the possible paths in a given target area. We design algorithms to clean and select raw taxi GPS data, extract features, cluster these selected data and visualize the clustering result to get the possible paths. Then we try to get the shape and direction information of these paths. All of our experiments are conducted on a taxi GPS data set collected form Beijing, China[1].

2 Introduction

Nowadays many very useful applications are based on Geographic Information Systems (GIS). One example could be vehicle route planning and navigation. The functionality of these applications highly depends on the completeness and correctness of the underlying maps. However, the maps we have today are suffering from many errors and insufficient information. How can we refine and update our maps? Since human resource is very costly these days, it's a good idea to develop some automatic methods to do this job.

Given that many vehicles have GPS equipment and most of them run along streets, why don't we make some smart use of these vehicle trajectories? We can imagine that these trajectories contain very rich map information. The question is how to mine the information out.

With sparsely sampled GPS point sequences of many different taxis, can we just draw the map out? As shown in Figure 1, by plotting together many GPS sequences, we can only get a very vague guess about the detail-complex overpass structure, not to mention drawing the map out. What is lacking here is the geometry and logic of the map—the possible paths in this area, the shape of these paths and the direction of these paths.

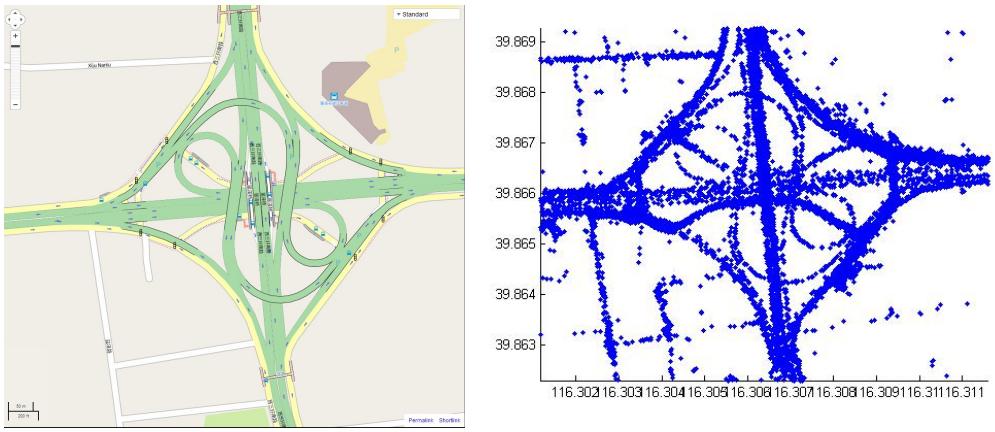


Figure 1: How to reconstruct detail-complex overpasses?

Figure 2: Roadmap of Learning Possible Paths

A complete map refining system will contain two steps. First, given a target area the system will learn about the map of this area from vehicle GPS trajectories. Second, The system will compare the learned map information and the existing map and then update the map basing on the comparison.

This project will provide a novel map learning algorithm for the first step of the map refining system. Because overpasses are a kind of most complex structures in maps, this article will take overpass reconstruction as an example to illustrate the algorithm. But we need to keep in mind that this algorithm is supposed to learn map information of any given area.

3 Problem Set Up

All the following materials will focus on reconstructing the overpass in a selected region (like in the left part of Figure 1) basing on sparsely samples GPS sequences (the right part of Figure 1 shows the overlapping of these sequences).

The first question we need to ask is what do we need to know about the overpass (or the map in this region). A good answer to this question would be all the possible paths that a vehicle might take in this region. Knowing that, we can draw the map. We know that each of the taxi GPS sequences going through this region necessarily takes one of these possible paths. Hence we can learn about these paths in two steps: 1) cluster the taxi sequences which share the same path together, and 2) learn about the shape and direction of these paths.

4 Learning Possible Paths

In this section we will try to cluster GPS sequences sharing the same path out.

The road map of the first stage of this project—learning possible paths—is shown in Figure 2.

In the first step data cleaning, we select out our desired data—those GPS sequences who go through the target area—and remove the dirty ones of them. The second step is feature extraction. Here we design three sets of features which separately corresponds to our baseline algorithm and two complementary algorithms. The first two steps are implemented with C#. The third step is clustering. Here we use a open sourced K-Means Clustering packet[2]. The final step is result visualization. Here we use Matlab to draw different clusters in different colors onto a plain panel. Then we can see the paths that we found.

4.1 Feature Extraction

Like in Figure 3, the density of the GPS points is very important to our reconstruction algorithm. If the GPS sequence of a single trajectory is sparse, like 60 seconds each in our data set, we will only have about 1 to 3 points on the overpass. It means a single GPS sequence in this region does not necessarily look like the path, although it is on the path. It also means two GPS sequences sharing the same path do not necessarily look similar. So first we need to design features that make GPS sequences sharing the same path look similar.

The idea is to make the point sequences denser by regulation and interpolation. Figure 4 shows our designed feature. By connecting GPS points by lines, we can get the approximate intersections

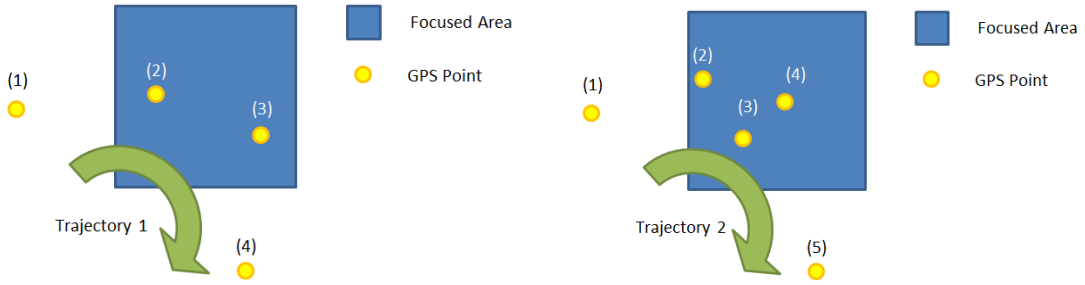


Figure 3: Example of Taxi Trajectory Sequences

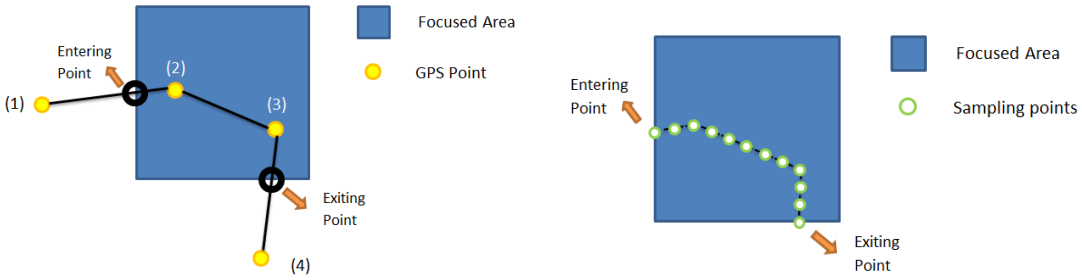


Figure 4: Feature Extraction

of the trajectory and the boundary of the target area. Then we get the trajectory segment between the entering point and exiting point and evenly sample this curve segment to get a certain number of intermediate points. These points, together with the entering point and exiting point, form the feature of our clustering algorithm. Through this process, we can expect our resulting point sequences look like the path and look similar if they share the same path.

Figure 5 plotted the extracted feature of all of the GPS sequences together. We can notice the hidden patterns in these features just by naked eyes. That proves to some extent the correctness of our way of feature extraction.

After we get the features of the GPS sequences, we can run K-Means algorithm or EM algorithm to cluster them. Each cluster in the result would represent a possible path in this region. By drawing out original GPS sequences in the same cluster, we can then learn the shape and direction of this path.

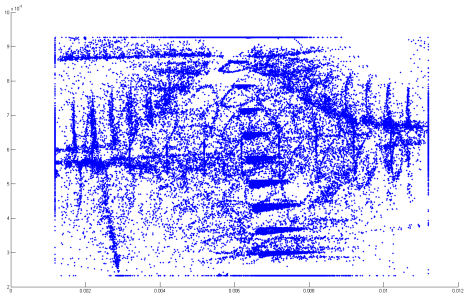
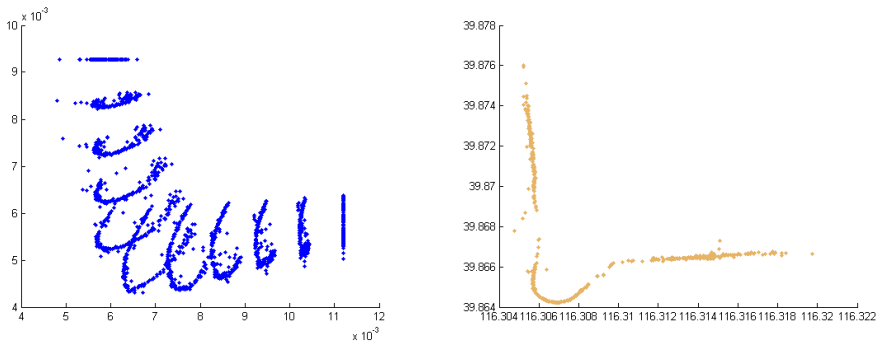


Figure 5: Extracted Features

4.2 Clustering Result and Analysis

Applying K-Means algorithm on the features in Figure 5, we can get clusters like in Figure 6(a) and Figure 6(c). Then we can trace back from features and original GPS sequences. By plotting GPS sequences in the same cluster together, we can get paths like in Figure 6(b) and Figure 6(d).

Each of the clusters in the clustering result represents a possible path in the target area. Figure



(a) Clustered Features Example A

(b) Clustered GPS Sequences Example A

(c) Clustered Features Example B

(d) Clustered GPS Sequences Example B

Figure 6: Two examples of clustered features and GPS sequences

6(a) and Figure 6(b) represents the path on the main road that enters the target area from the north and then takes a left turn. Figure 6(c) and Figure 6(d) represents the path on the side road that enters the target area from the south and then exits from the north.

Figure 7 shows all the clustered GPS sequences, each color representing one possible path. We can see that our algorithm learned the geometry and logic structure of this very complex overpass. We have 32 clusters in total in this example. 16 for the main roads and 16 for the side roads (One can turn right, go straight, turn right or make a U turn when entering the intersection from one direction. There are four directions to enter for both the main roads and the side roads).

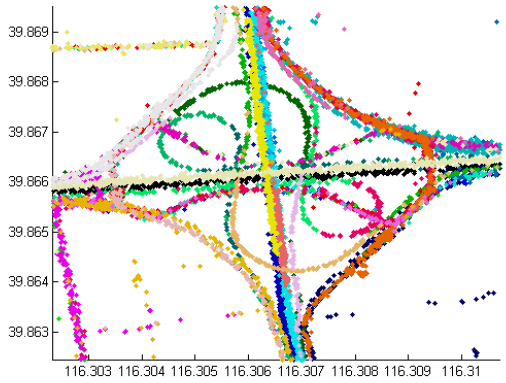


Figure 7: Clustered GPS Sequences

5 Learning Path Shape and Direction

Now we have had a idea about what are all the possible paths in the target area. By plotting all the GPS sequences belonging to the same path together, we can know about how the path goes by observation. But we still haven't extracted the shape and direction of the paths from the original GPS data.

The shape and direction of a certain path can be represented by a sequence of points. There are two factors we need to know about this sequence of points—the position of these points and the order of these points.

Like in Figure 8, the easiest way to learn about the shape and direction of the paths is by using the cluster centers we learned at the last clustering stage. But there are two problems in this approach. First, since we tried to approximate path curves by straight line segments in the feature design stage, our cluster centers would suffer from shifting from the original path curves. Second, the number of points in the cluster centers are decided by the need of the clustering stage. It is the result of the trade off between clustering performance and running time. Now we might want more points to better represent the curve of the path. The points in a cluster center might not be enough. These two reasons explains why the reconstructed overpass in Figure 8 looks so bad.

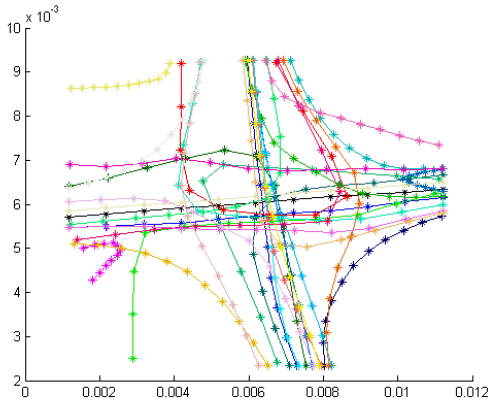


Figure 8: Cluster Centers

So we need an additional algorithm to learn about the shape and direction of the paths. This algorithm has two steps. The first step is to get the collection of points by applying K-Means algorithm on the GPS points in the same cluster to represent a certain path. Since we get these points basing on the original GPS data rather the designed feature, these points would have no shifting problem. The second step is to learn the order of these points. Each point now is the center of a GPS points cluster. Since we have the order information of the original GPS points, we can set up a partial order relationship between the shape points basing on the order relationship of the original GPS points. Then after we get the order of these points, we would know complete shape and direction information of the paths. Figure 9 shows examples of the reconstructed paths.

6 Conclusion and Future Work

This project developed an algorithm to learn about map information from sparse GPS sequences. The input of this algorithm is a target area and a GPS sequence data set. The output of this algorithm is the shape and direction of the possible paths in the target area.

Future work of this project would be developing algorithms to compare our learned map information with the existing map and thus refine and update the existing map.

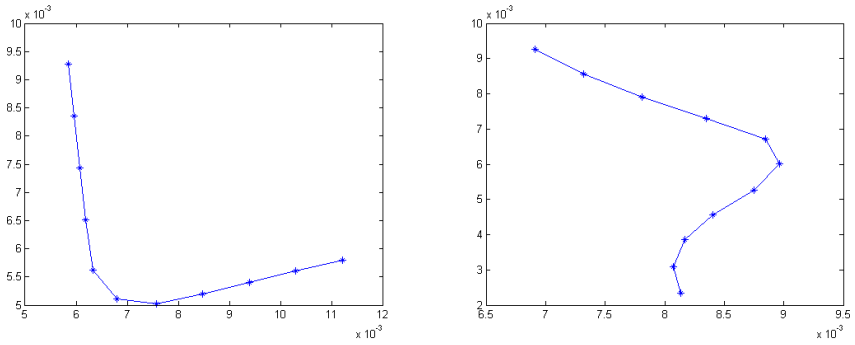


Figure 9: Example of Reconstructed Path

7 References

-[1] The Taxi GPS Sequence data set is downloaded from http://sensor.ee.tsinghua.edu.cn/
-[2] The open sourced K-Means clustering executable is downloaded from http://mercury.webster.edu/aleshunas/Source%20Code%20and%20Executables/Source%20Code%20and%20Executables.html