Traffic Predicion With Advanced Graph Neural Networks

Mid-term Report

11812804 董　正

11813225 王宇辰

11811305 崔俞崧

Supervisior: 宋轩

Department of Computer Science and Engineering

Dec. 2020

Content

1 Preliminaries

1.1 Review

1.1.1 TTE

Travel Time Estimation (TTE) is one of the most important researching topic in the traffic forecasting field. Estimating the travel time of any path in a city is of great importance to traffic monitoring, route planning, ridesharing, taxi dispatching, etc. On Sep. 2020, DeepMind published a blog named *Traffic prediction with advanced Graph Neural Networks*. This blog briefly described the whole industrial structure of estimated times of arrival (ETAs) techniques applied in Google Map but did not given any detailed implementation or any code. Our work is based on the model structure of TTE proposed in the blog.

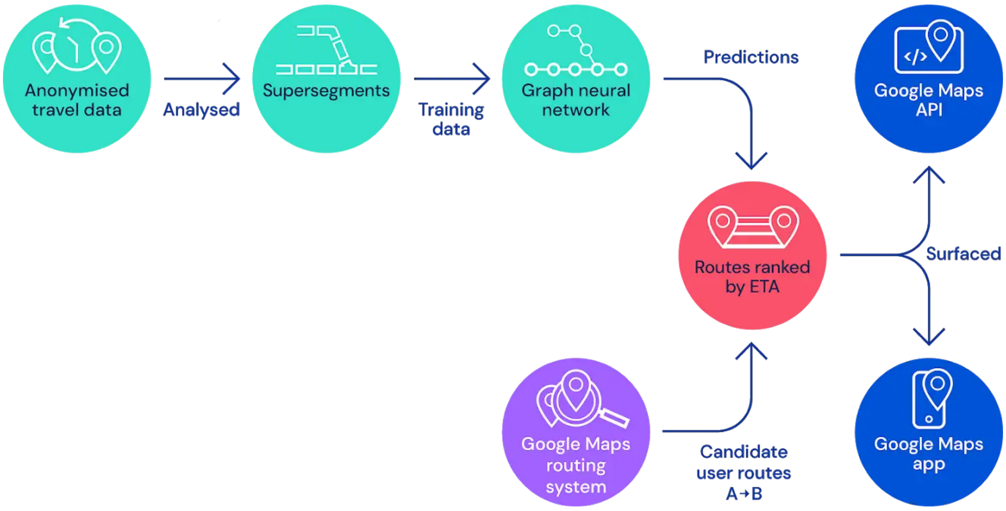


Figure 1: Architecture

1.1.2 Goal

Our ultimate goal (tentative) is to implement the industrial structure and apply it to the open source databases in China, then compare the performance with the state-of-the-art structures and find its application value. This semester, we will first learn about the relative algorithms. More specifically, the two main algorithm we will use: TTE algorithm and Graph Neural Network (GNN). After that, we will learn the methods of processing traffic data. Finally, we combine the algorithms and apply them on some open source database.

1.2 Introduction

In the last stage, we stated the basic ideas of two state-of-the-art algorithms named *DeepTTE* and *DCRNN*. As for this month, we have got some open source data and learned how to process the data. What's more, we got deep in the code of *DeepTTE* and researched on a new concept called **Travel Time Index (TTI)**. And we also researched on the *Supersegment* and proposed a new computing process.

Breifly, we will state our work in this report as:

- Implementation of *DeepTTE* by 王宇辰
- Travel Time Index by 崔俞崧
- A Research on Supersegment by 董正

2 Implementation of DeepTTE

Three parts of implementation is mentioned below. First is dataset, gives the basic information of data composition to analysis. Second is algorithm, gives the structure and algorithm in the procedure of implement. Third is result, gives loss after training and loss in testing. [1]

2.1 Dataset

Basic infos driverID, dateID (the date in a month, from 0 to 30), weekID (the day of week, from 0 to 6 (Mon to Sun)), timeID (the ID of the start time (in minute), from 0 to 1439), dist (total distance of the path (KM)), time (total travel time (min), i.e., the ground truth. You can set it as any value during the test phase) Trip info for each node in one time:

- states: the sequence of taxi states (available/unavaible)
- lngs: the sequence of longitutes of all sampled GPS points
- lats: the sequence of latitudes of all sampled GPS points
- time_gap: Each value indicates the time gap from current point to the firt point
- dist_gap: Each value indicates the dist gap from current point to the firt point

Statistic info as mean value and standard deviation of trip infos.

2.2 Algorithm

The alogrithm is applied with Pytorch. Using ConvolutionNN and RecurrentNN structures. In Attribute Component only mix personal info, without use complex models. In Spatio-Temporal Component, it is dived into two parts, first is the Geo-Conv layer using ConvolutionNN model, second is beyond the layer, using LSTM model, a kind of RecurrentNN. In Multi-Task Component, mainly using error functions to estimate the model. The Entire Estimator uses rooted mean squared error and mean absolute error, and the Local Estimator uses mean absolute percentage error and only apllied in training. Also, in training, the two estimator is weighted combined.

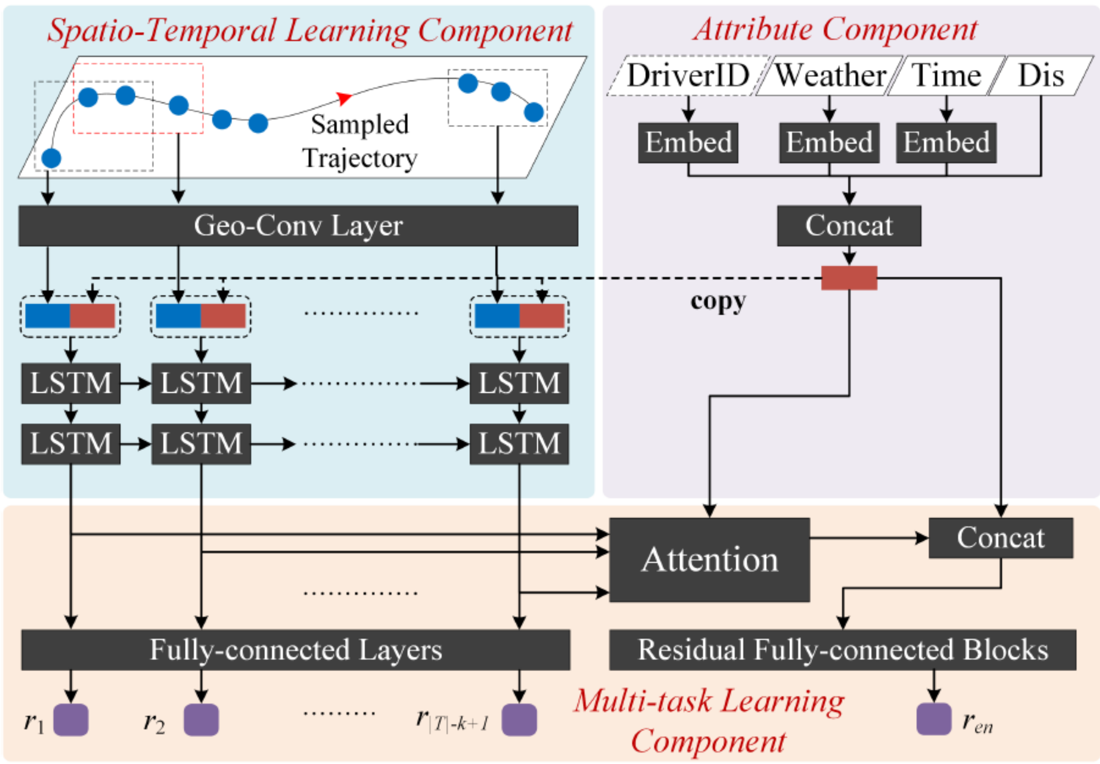


Figure 2: Model

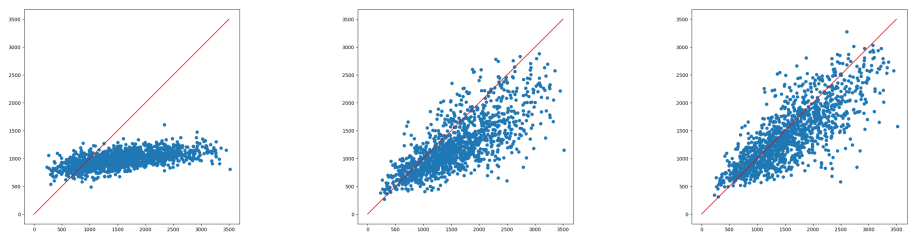
2.3 Result

The result contains two columns, the first is real time of time travel, the second is predict time of time travel.

```
3074.000000 2794.380371
```

```
1654.000000 1769.522095
2792.000000 2441.029297
1401.000000 2544.872070
2609.000000 2683.422852
2483.000000 2852.733643
2217.000000 1989.398804
1342.000000 1352.451904
2067.000000 2509.079102
1747.000000 1898.822144
2044.000000 1555.407837
1764.000000 1918.074463
1917.000000 1928.597168
2247.000000 1921.359863
1373.000000 1455.389038
2659.000000 2034.604614
2866.000000 2539.196777
2049.000000 1863.575684
2494.000000 2103.954102
1987.000000 1525.647583
3074.000000 3030.991699
...
```

The training contains 100 epoches.



(a) Epoch 1 (b) Epoch 5 (c) Epoch 100

Figure 3: Training Epochs

From above, the result is close to the function $y = x$, shows that the result is accurate to estimate the real time. Also, with line chart below, we can see that, error in training is rather low, about 0.02 after training, and testing on the test set, the error is about 0.2, it is a good result in travel time estimation.

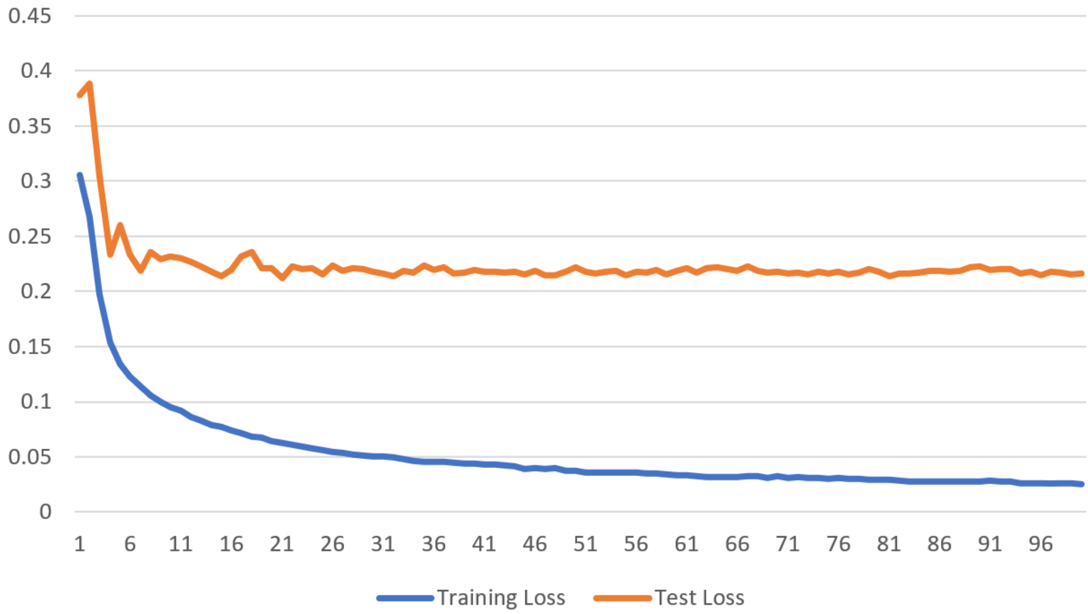


Figure 4: Loss

3 Travel Time Index

3.1 Introduction

The **TTI (Travel Time Index)** industry's most used evaluation index of urban congestion degree is the ratio of the actual travel time and the free flow time. The larger the value, the worse the traffic operation status, and the congestion level is generally positive. Related, other abnormal weather conditions (such as rain, snow, fog, etc.) or abnormal road conditions may also affect the value of TTI.

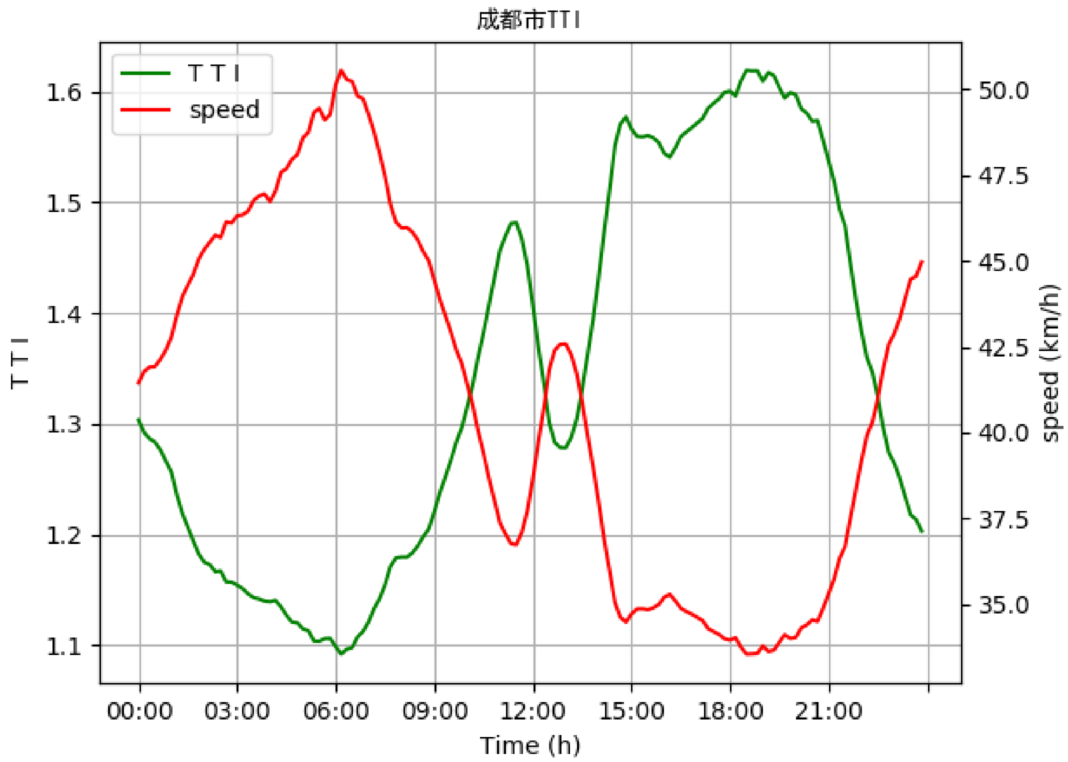


Figure 5: TTI Data Curves of Chengdu

3.2 Basic Idea

The basic idea of Speed: If a link has two time slices, t_1 and t_2, and the link's length is S, then the average speed v of the link is: $v = 2S/(t_1 + t_2)$ during the period from t_1 to t_2.

The basic idea of TTI: In the same link in a time slice, $TTI = free\ flow\ speed/actual\ speed$.

3.3 Calculation

$$TTI = \frac{\sum_{i=1}^{N} \frac{L_i}{V_i} \cdot W_i}{\sum_{i=1}^{N} \frac{L_i}{V_{free\_i}} \cdot W_i}$$

- L_i: length of the road
- W_i: weight of road
- V_i: real time traffic information
- V_{free_i}: free flow velocity

3.4 Motivation

The reasons for TTI calculation are as follows:
DeepTTE algorithm has some shortcomings:

1. Excessively strict requirements on data. The input of DeepTTE algorithm is trajectory data, which is usually unstable, the requirements of the algorithm are very strict. In the data preprocessing stage, the interval point of trajectory data should be between 200-400m and should not be too large. In addition, the data used in the demonstration of the paper are all selected and processed.

2. The effect is not stable enough. There are various problems such as excessive error and data loss in trajectory data, and the results obtained by using such data for training will be too deviated from the ideal.

3. There is a certain distance from academia to industry. In industrial applications, the trajectory data obtained is quite different from the data set used in this paper, and the results obtained when directly used as input are not ideal due to the lack of data and errors. The purpose of TTI calculation is to improve DeepTTE algorithm and change the input data from the original track data to the road TTI data. TTI data has the advantages of more stable and smaller error. Since TTI data is calculated through the trajectory data, the calculation process is the first step of the processing of the trajectory data, so that the TTI data has a smaller error than the trajectory data.

3.5 Dataset

- Data source: Didi Chuxing GAIA Intiative
- Region: Chengdu

- Time: 2018-10-01 to 2018-12-01
- Content:
  - GPS track of taxis with timestamps
  - Coordinate of road and district boundaries
  - TTI and average speed of districts

轨迹数据

字段	类型	实例	备注
司机ID	String	glox.jrrlltBMvCh8nxqktdr2dtopmlH	已经脱敏处理
时间戳	String	1501584540	unix时间戳，单位为秒
经度	String	104.04392	GCJ-02坐标系
纬度	String	104.04392	GCJ-02坐标系

city_district,road文件说明

字段	说明	样例数据
obj_id	TTI对象ID	841
batch_time	时间	2018-01-01 00:00:00
tti	交通指数数据	1.18665
speed	平均速度	47.3983

boundary文件说明

字段	说明	样例数据
obj_id	TTI对象ID	283203
obj_name	TTI对象名称	百日红中路:三环路,银杏大道
geom	TTI对象几何范围	MULTILINESTRING((104.13743 30.60591,104.13832 30.60538),(104.13832 30.60538,104.13843 30.60532),(104.13843 30.60532,104.13855 30.60525),(104.13855 30.60525,104.13921 30.6049,104.13984 30.60454)...)

Figure 6: Dataset

3.6 Data Processing and TTI Computing Method

It can be seen from the calculation formula of TTI that the key to calculating TTI is to get each feature in the following table, which are road network data, real-time traffic data, free flow data and weight data.

The road network data and real-time traffic data can be obtained directly from the data set, while the free flow data and weight data need to be extracted from the data set by using some methods. The extraction method of free flow data is as follows: since free flow is defined as the traffic flow information when there is no congestion, the traffic flow information located at 2:00-5:00 am can be selected as the free flow information since this is the time when congestion is least likely to occur. Extraction method of weight data: the number of vehicles passing through each section in a month is counted and recorded, and this number is the weight data. After processing the data information, the TTI calculation formula can be used to calculate the TTI data of each road.

Feature	**Detail**
Road network data	Basic map road network data, including road shape and unique id
Realtime Traffic data	Relying on dripping massive floating car data, real-time traffic data released by Didi
Free flow data	From the Drip Trajectory database, the road speed generated by the excavation
Weight data	The total number of vehicles passing the road within a natural month

4 Supersegment

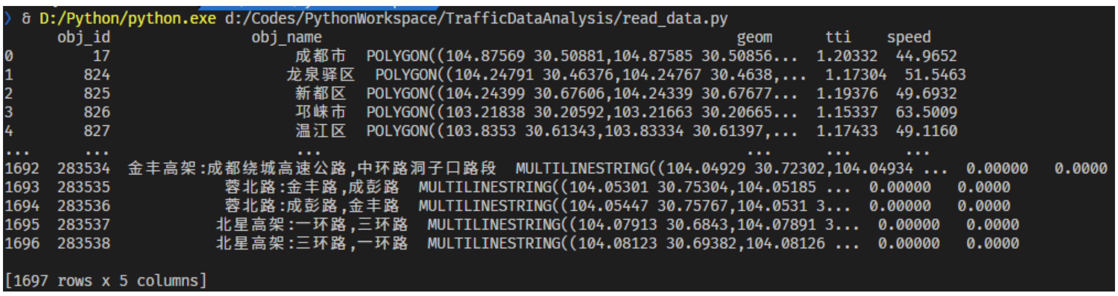
4.1 Introduction

Supersegment is a new concept proposed in a blog by *DeepMind*. But unfortunately, *DeepMind* did not give any formal definition. Instead, they stated that "We divided road networks into 'Supersegments' consisting of multiple adjacent segments of road that share significant traffic volume." [2]

As for my understanding, Supersegment is a division of road that the adjacent segments have something in common such as speed and volume, but at the same time, there are also differences that big enough to distinguish them.

4.2 Dataset

- Data source: Didi Chuxing GAIA Intiative
- Region: Chengdu
- Time: 2018-10-01 to 2018-12-01
- Content:
  - GPS track of taxis with timestamps
  - Coordinate of road and district boundaries
  - TTI and average speed of districts



```
❯ & D:/Python/python.exe d:/Codes/PythonWorkspace/TrafficDataAnalysis/read_data.py
     obj_id                 obj_name                                              geom      tti    speed
0        17                      成都市  POLYGON((104.87569 30.50881,104.87585 30.50856...  1.20332  44.9652
1       824                     龙泉驿区  POLYGON((104.24791 30.46376,104.24767 30.4638,...  1.17304  51.5463
2       825                      新都区  POLYGON((104.24399 30.67606,104.24339 30.67677...  1.19376  49.6932
3       826                      邛崃市  POLYGON((103.21838 30.20592,103.21663 30.20665...  1.15337  63.5009
4       827                      温江区  POLYGON((103.8353 30.61343,103.83334 30.61397,...  1.17433  49.1160
...     ...                      ...                                               ...      ...      ...
1692 283534 金丰高架:成都绕城高速公路,中环路洞子口路段  MULTILINESTRING((104.04929 30.72302,104.04934 ...  0.00000   0.0000
1693 283535            蓉北路:金丰路,成彭路  MULTILINESTRING((104.05301 30.75304,104.05185 ...  0.00000   0.0000
1694 283536            蓉北路:成彭路,金丰路  MULTILINESTRING((104.05447 30.75767,104.0531 3...  0.00000   0.0000
1695 283537           北星高架:一环路,三环路  MULTILINESTRING((104.07913 30.6843,104.07891 3...  0.00000   0.0000
1696 283538           北星高架:三环路,一环路  MULTILINESTRING((104.08123 30.69382,104.08126 ...  0.00000   0.0000

[1697 rows x 5 columns]
```

Figure 7: Data

4.3 Problem Description

Currently, there is no algorithm of Supersegment or "road division". Therefore, I need to work on myself.

Now, we have

- Coordinate of road boundaries
- GPS data of taxis
- Timestamps

And we need to work out a partition of a road.

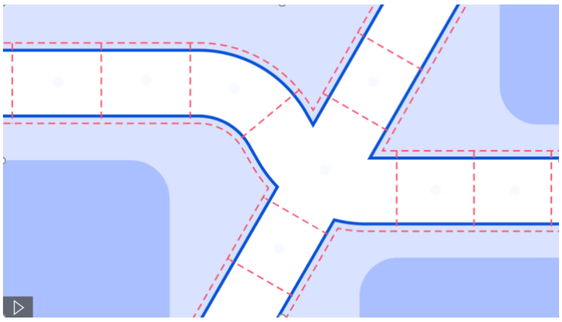


Figure 8: Supersegments

4.4 Model Design

In this section, I will propose a model to compute Supersegments. At the very first, I need to ensure that I will base on **speed** to do the division. There is a three-step approach of my design:

1. Locate GPS coordinates of taxis into corresponding roads
2. Use the timestamps to calculate average speed and regead it as the instantaneous speed of midpoint
3. Apply clustering algorithm to these midpoints

First, since we know the boundaries of the roads, we can determine the coordinate is on which road. After that, we draw the GPS points on the road. Because we know the timestamp of each point, we can calculate the average speed of two adjacent points and just consider it as the speed of their midpoint. Note that the closer the points are, the closer the average speed is to the actual instantaneous speed. So the best way is to select adjacent points if our data is abundant.

Next step is to apply clustering algorithm like *K-Means* to find a certain number of clusters of the midpoints. Every midpoint has three features:

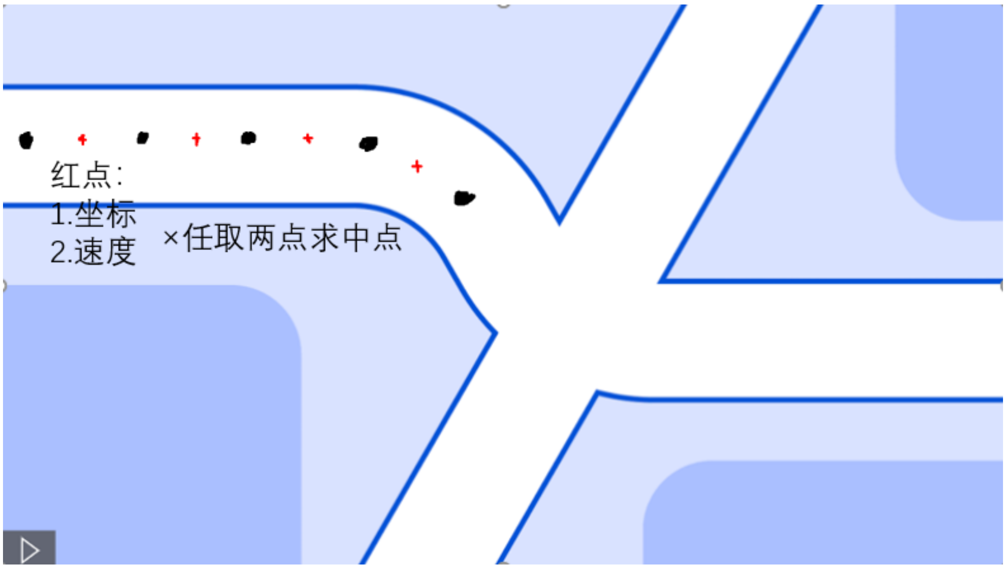
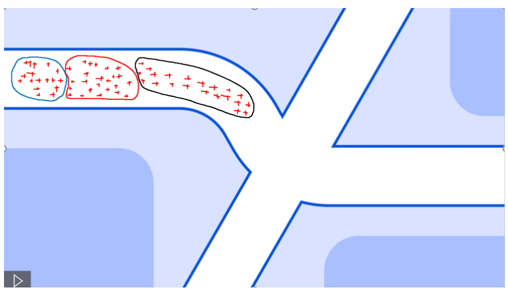


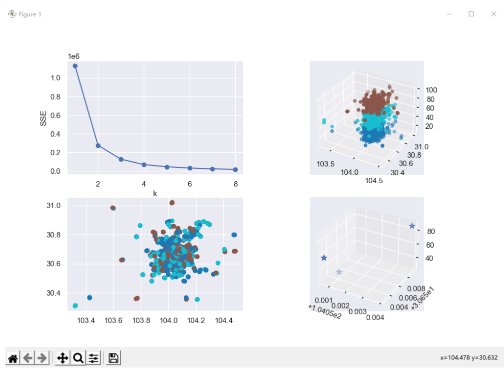
Figure 9: Find the midpoints

- longitude
- latitude
- speed

So actually this is a three-dimension clustering, but the result we need is two-dimensional, which means we need to balance the weight of these features carefully to get a correct partition. And finally, these clusters give us segments.



(a) Clusters



(b) K-Means Example

4.5 Conclusion

While it is just my personal thought, there are still many problems that are hard to handle. One vital task is how to evaluate a Supersegment division is good. I do not have any idea. And I still do not know whether *K-Means* algorithm will work well on this topic. But anyway, I think my model is at least feasible and I will continue researching on it if given more time.

References

[1] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *AAAI 2018*, January 2018.

[2] O. Lange and L. Perez, "Traffic prediction with advanced graph neural networks." https://deepmind.com/blog/article/traffic-prediction-with-advanced-graph-neural-networks, Sept. 2020.