Design and Implementation of A TTE System

Final Report

11812804 董　正

11813225 王宇辰

11811305 崔俞崧

Supervisior: 宋轩

Department of Computer Science and Engineering

June 2021

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.

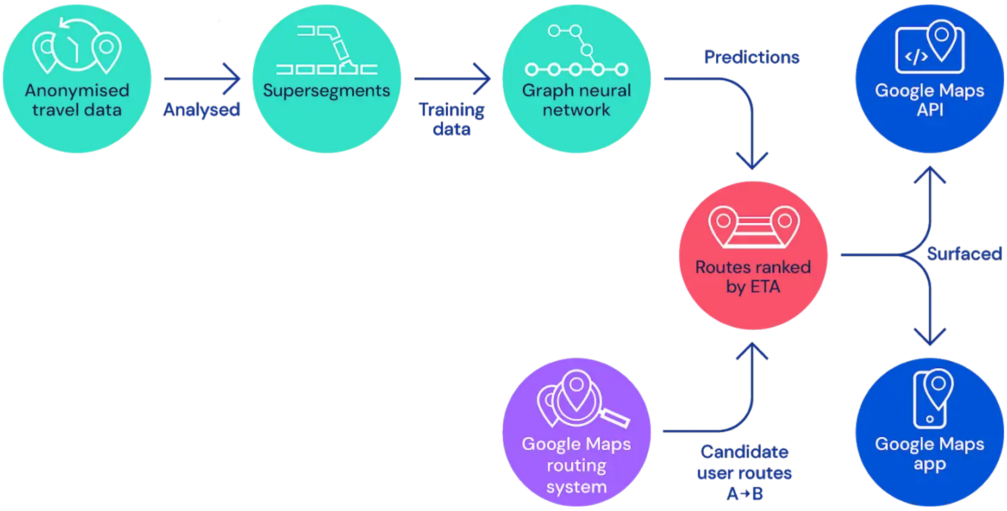


Figure 1: Architecture

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 implement a TTE system base on the work we done in the last term, combining *Supersegment* and TTI. We will try to work out an interactive application with graphical user interface.

1.2 Introduction

In last stage, we completed the first version of our TTE web application.
Breifly, we will state our work in this report as

- UI Redesign and Simple DB for TTE Web APP 董正 & 王宇辰
- TTI Optimization by 崔俞崧

2 UI Redesign and Simple DB for TTE Web APP

2.1 Introduction

Our application is a front-back web system.

Front end:

- *Leaflet*: A JavaScript map library
- 高德地图: Provider of the base map
- *jQuery*: A JavaScript communicator

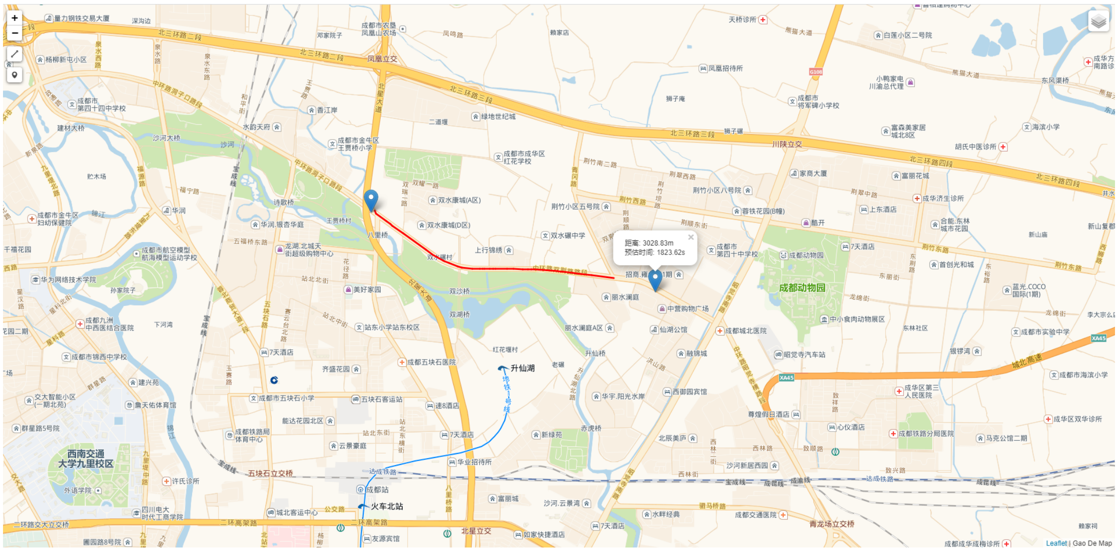
Back end:

- *Flask*: A *Python* back-end framework
- *NetworkX*: A *Python* library for graph analysis
- *GeoJSON*: A special format of JSON that represents geo data

2.2 System Overview

1. Convert orginal road data to *shapefile* format
2. Build geo-graph with *NetworkX*
3. Select start and end points in web front, use *jQuery* to send request to back-end server
4. Run *Dijkstra* alogrithm in the graph to find shortest path and distance and store them in database
5. Query database for the result
6. Send *GeoJSON* data to front-end and show the calculated path

7. Calculate TTI, and use it to work out the estimated time of arrival
8. Send the final result to front-end



距离: 3028.83m
预估时间: 1823.62s

Figure 2: Web Application

2.3 Database Design

For the ease of use, we designed a simple database with two tables.

Table 1:

- Coordinate of the source point
- Coordinate of the target point
- Distance
- Path (a series of points)

Table 2:

- Coordinate of the source point
- Coordinate of the target point
- Edge (contains road infomation)

Database Interaction:

1. Input: Coordinates of the source point and target point
2. Calculate the minimum of sum of manhattan distance to get the nearest points
3. Query **Table 1** to get distance and path
4. Use the path to query **Table 2** iteratively to get the roads
5. Return distance and roads

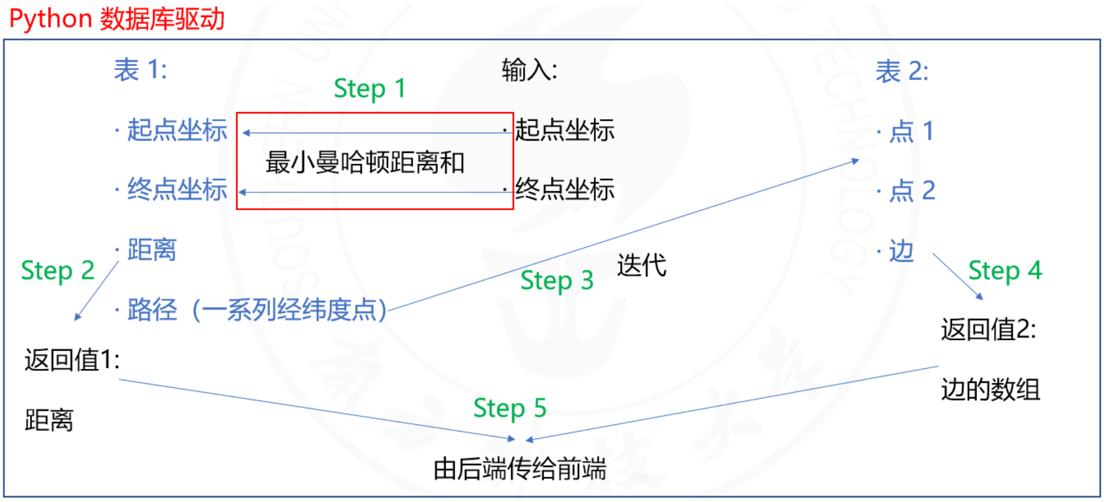


Figure 3: Database Query

3 TTI Optimization

3.1 Purpose Of TTI Optimization

The main reason for TTI optimization is that several deficiencies of the current TTI calculation results are found in the actual application process, and the prediction accuracy could be improved by optimizing the current results. First of all, when calculating unpopular road sections, it is easy to have no track records, which leads to null TTI value. Secondly, when the time granularity is finely divided, the adjacent values of TTI fluctuate greatly, and the results show a zigzag pattern, which will lead to large prediction errors. Finally, the amount of data currently utilized is insufficient, resulting in the failure to utilize all the data effectively in the calculation. Data should be filtered reasonably to make full use of the data.

3.2 TTI Optimization Methods

1. Complete TTI null values through interpolation. Due to the time limitation of adjacent TTI values, linear interpolation can be used when the null value is small. However, when the null value is large, the simple interpolation method cannot predict the TTI value of the void, and other ways are needed to complete it.

2. Filtering is carried out after completion. Since the amount of data is not large, when the time granularity is fine, it is likely that the number of vehicles in each period is small, and the speed of different vehicles will vary greatly, so filtering is needed to reduce the impact caused by fluctuations and improve the accuracy.

3. After interpolation, MAE, variance and other indicators are used to evaluate various filtering methods and select the best filtering method. MAE can be used to evaluate the distortion of TTI before and after filtering, while variance can be used to evaluate the effect of filtering.

3.3 TTI Optimization Process

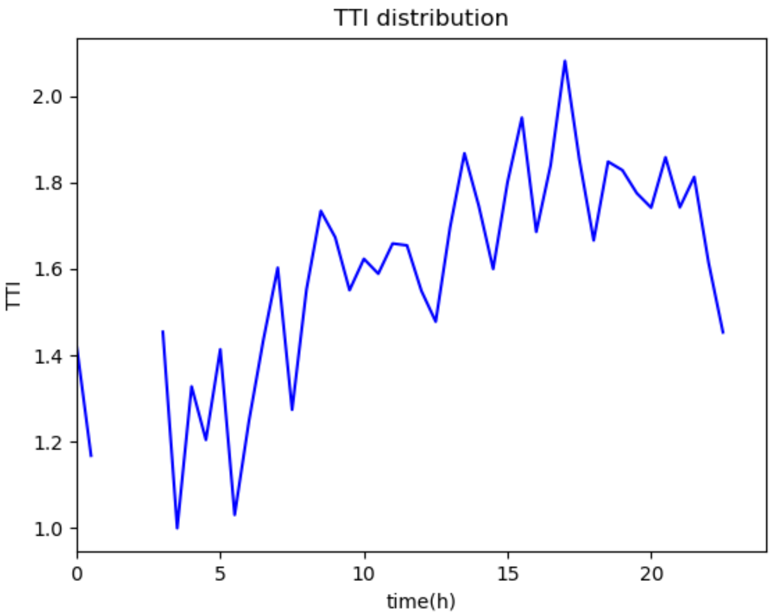


Figure 4: TTI Without Processing

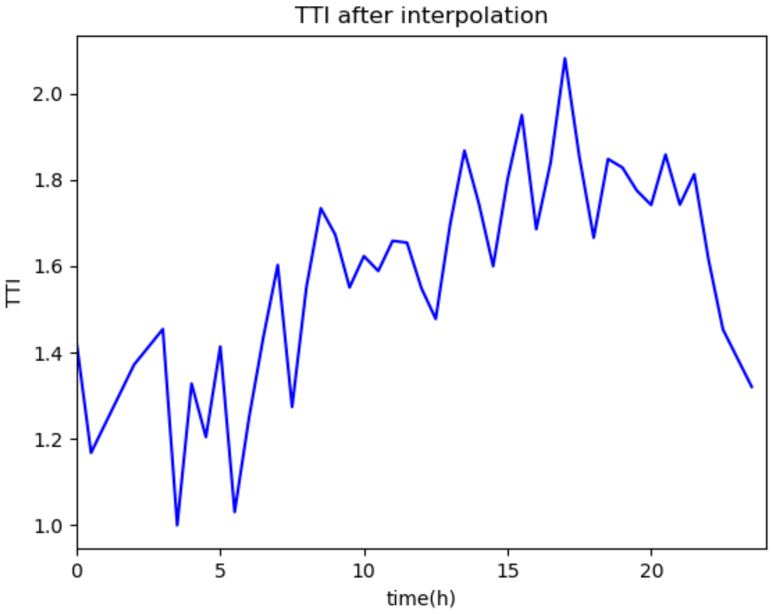


Figure 5: TTI After Linear Interpolation

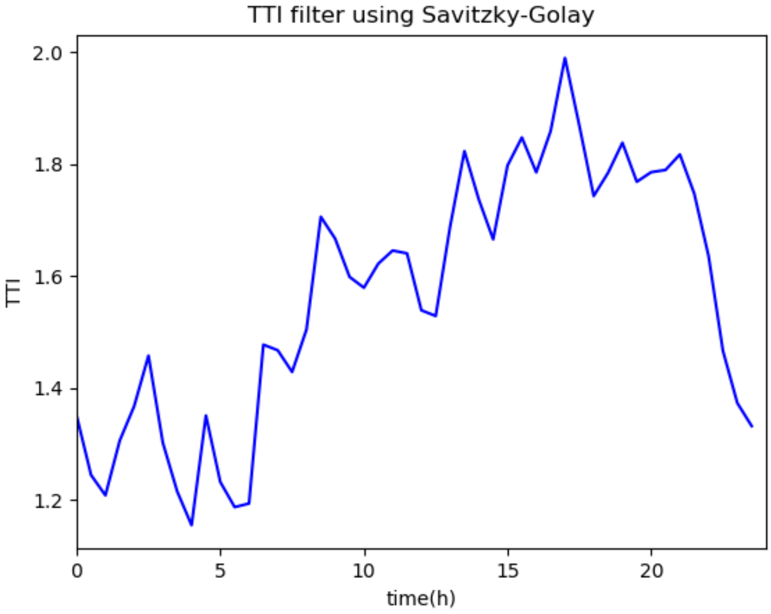


Figure 6: TTI Filtering Using Savitzky-Golay

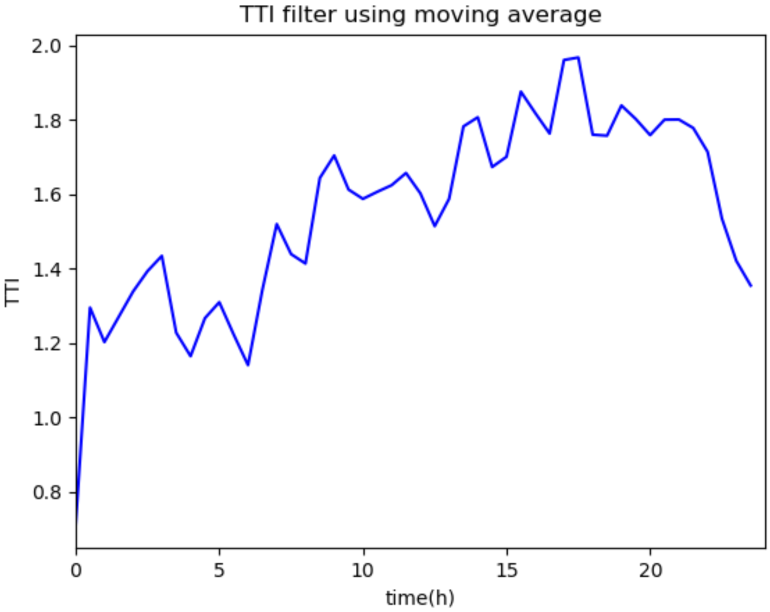


Figure 7: TTI Filtering Using Moving Average

3.4 Result Analysis

1. It can be seen from the processing process of 3.3 that interpolation and filtering can fix the issue of null value and large fluctuation of TTI.

2. The filtering results obtained by different filtering methods are different to some extent. A more appropriate filtering method can be obtained through comprehensive evaluation of various evaluation indexes.

3. There are situations where the TTI result has many null values, and interpolation and filtering methods cannot be used at this time, and data processing should be considered.

4. In order to ensure the efficiency of the algorithm, there is a certain error in the calculation process, and the processing of the results cannot eliminate the error in the calculation process.

3.5 Ascending Direction

1. Increase the amount of data and improve the efficiency of the algorithm, so that more data can be used to build the model and the algorithm can better deal with special cases.

2. Since there are a large number of tracks from 8 am to 8 pm, and people are more likely to query the travel time during this period when using the app, this part of the time can be used instead of the using of whole time.

3. When analyzing the TTI image, it is found that the TTI does not change much in the early morning of each day, and it is also the time when the null value and severe fluctuation are most likely to occur. Consider combining these periods to calculate the TTI.

4 Conclusion

In this semester, we successfully builded a complete web application having front-end, back-end, and database. Still, its accuracy needs to be improved. However, now it is ready for deployment and usage. Therefore, we achieved our goal proposed in the first report.

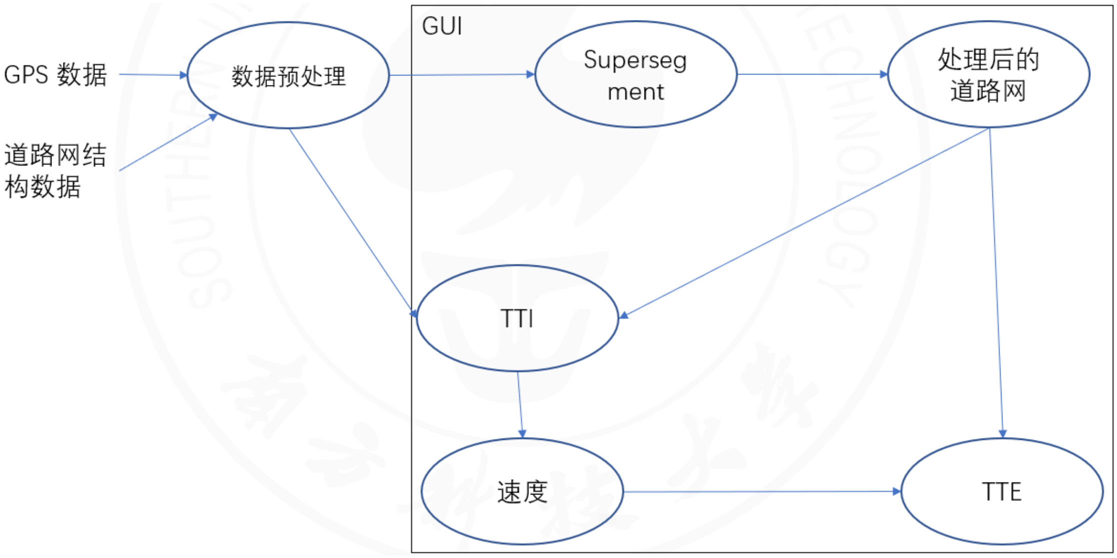


Figure 8: TTE Web Application Structure