

Ui Design:Data analysing and visualization of Hanzhou Metro Traffic

Haotian Xue 518021910506

Report of CS241 Final Project

1 Introduction

As is mentioned in the project introduction, these years is witnessing a surge of urban metro traffic in many places in China, which has been facilitating our life and at the same time, causing overcrowding problems. And that makes it necessary to make analysis of the metro traffic data from our daily life.

In this project, I made a user interface based on Qt5 to process the data from the AFC dataset and implemented some user-based functions to benefit the users. The main functions include: downloading necessary data for the users, make a visualization of the in-out flows and make route planning based on the user's requests.

Other work includes: optimzing route advicing algorithm based on some prior data, making visualization of the crowding degree of stations, lines and integrating some Hangzhou elements in the user interface and so on.

2 Implement details

2.1 Developer environmnet

All of this project are written in Qt5(C++) in Linux(ubuntu 18.04). Other complilation environmnets include: Linux g++, sqlite and qmake.

2.2 Ui structure

The Ui are made up of four main components: open window, dataloader, visualization window and route planning window. The dataloader can scratch the data from the raw data file and insert them into a sqlite database, which is then used by the other functional windows. All of the four parts interact with each other and all of the three windows interact with the users. The structure details are shown in Figure1.

2.3 Funtion implementation

2.3.1 Dataloader and filter

The dataloader aims the move data in the raw .csv file into a data container which can be then used to serve the data analysis.

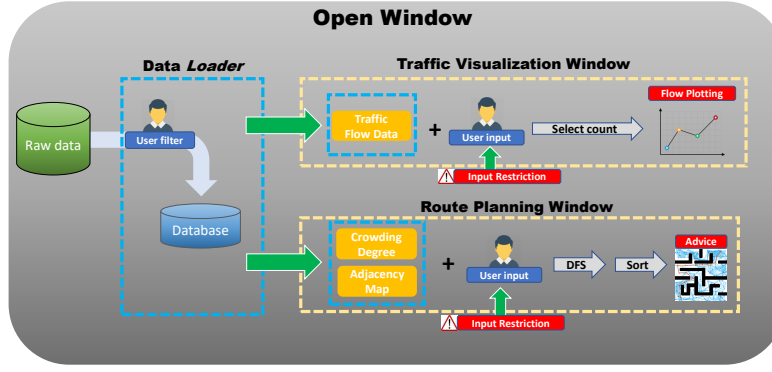


Figure 1: Ui Structure

At first, I try to simply save the data into a container written in the memory of the computer. This actually makes the data processing really fast, however, this process will make up a fair share of memory in my computer and will cause the computer to run out of memory if it is not carefully designed.

So I choose Sqlite3 database as a container of these raw data, which from my knowledge can be more efficient especially on data selection.

Data filter is a kind of interaction with the users in data processing, users can make selections of their areas of interest to make downloading. It should be noted that:

1. As some data is necessary for the data analysis, I made some little restrictions on the selection.
2. The dataloader works in a new thread to make sure that you can do other work when waiting.
3. Some functional windows can be opened if and only if when the data processing is done.
4. The message box will make suggestion for a proper operation.

2.3.2 Route planning

Route planning window is created to make route advice for users. The users can input their departure station number and their destination station number and get a optimized travelling route. I use DFS to make a search in the adjancencu map, and save all the possible route. For each possible, a score is calculated based on the following equation:

$$Score[i] = w_l L[i] + w_a A[i] + w_c C[i] (i = 1, 2, \dots, n) \quad (1)$$

where $L[i]$, $A[i]$, $C[i]$ are the length, station changing numbers and the crowding degree of the i th possible route. This is base on the everyday feeling: the longer way and more station changing will make the journey time consuming, the more crowding station is always not preferred. w_l , w_a , w_c are the coefficeints of the three factors, which represent the important of each factor. I set them to 0.5, 0.3, 0.2 respectively.

It should be noted that the calculation of length is based on the simply assumption that $L(i, j)$ represents the distance between station i and station j):

$$L(i, j) > L(i', j') (|i - j| > |i' - j'|) \quad (2)$$

So the length of the route can be computed as (n_i is the station number in route i , $L[i][j]$ is the number

of the j -th station in route i):

$$L[i] = \sum_1^{j=n_i-1} \left| L[i][j+1] - L[i][j] \right| \quad (3)$$

The crowding degree is calculated simply based on the prior data in the database, and $C[i] = n_i$.

2.3.3 Visualization of metro traffic flow

This part is designed to make the data more touchable for the users.

3 Results

4 Discussion