Report about the Application for Analysing Taxi Statistic Data

519021911182 Jieyi Zhang

1 Introduction

1.1 Background

With the development of hardware and the large application of big data, more and more sensors emerge around people, which generate a large amount of statistic data. To mine the value behind data and to support users with better service, there is an urgent need to realize the visualization of statistic data. Therefore, I develop a software application for data analyst to process and visualize the taxi statistic data in a more convenient way.

1.2 Purpose

In my application, I implement 4 fucntions as followed:

1.users could choose the time range of data to load. In addtion, users could see the progress and could cancel loading progress in half way.

2.The application could display the spatio-temporal demand patterns, whose parameters including time range, time step as well as display form could be tuned by users. Besides, the demand heat map is available if needed.

3.The application could show other value, including the distribution and the density of users' travel time and the distribution and the density of order fees, whose parameter could also be tuned.

4.The application offer users the navigation service. Users could set the start point and the end point, and see the navigation route.

2 Implementaion Details

2.1 Main Modules

In order to implement the goal mentioned, I code in Qt and use several libraries including SQL database, multi-threading modules, chart modules and MapViewer modules.

2.1.1 SQLite Database

To load data in *.csv*, I use the SQLite database module. The data users selected will be loaded in a SQLite database, so that the application could search the data needed in an efficient way. In the searching and counting part, I use *SELECT* function to search the data needed.

2.1.2 Multi-threading Module

Loading data may last a long time, and users may want to load a smaller data range since couldn't tolerate the long period. Thus, I use a multi-threading module to enable users to cancel the loading process. The loading process runs in the other thread, enabling the main thread to emit a signal to terminate the loading process.

2.1.3 Chart Module

To visualize the selected data, a chart module is essential. So, I choose the Qchart module to plot statistical chart including line chart, histogram and pie chart. With using the spline class in the Qchart module, the line ploted is smooth enough even the data point is more than little.

2.1.4 MapViewer Module

For purpose of displaying heat map and navigation, I take advantage of the MapViewer module, which including a inbuilt navigation API, allowing me to implement the navigation part.

2.2 Advantages

2.2.1 Data Comparison

In my application, users could choose 5 different series of data correspoding to 5 different regions at most. And the 5 differnt data series could display in the chart part in the meantime, so that users could not only see the trend of one regions but also compare data features in different regions.

2.2.2 Effcient Display

On account of the comparison of data, the application have to process multiple data series, so as to that the time cost will increase in direct proportion to the number of data series. Therefore, the users experience may decrease a lot if there is no optimizing process. At the beginning, I loaded all the data into a table. Naturally, it cost the application too much time for searching requiring data. Then, I loading the data into many tables according to the day the data belong to. It allow me to search data in smaller table accurately, which will reduce the time cost. However, after the optimizing process, the time cost is yet too long. So, I continue to divide the loaded data. I separate the data according to the region. So far, I create a hundred tables per day. These preprocessing will embed in the loading part, so the loading time may increase a little, while the time cost of ploting stage will decrease a lot. In the *Result* part, I will show how efficient the application is, comparing to the non-optimized version.

2.2.3 Heat Map with Obvious Heat Difference

There are 10 levels of heat in the heat map. It's crucial to set a standard to divide different heat level. No matter the difference of heat gradient is too large or too small, the display effect of the heat map is poor. At first, I choose the maximum of number of regions' orders as the level 10. And the gradient of a level is one tenth of the maximum. Nevertheless, the display effect is not good enough. After several attempts, I found that the threshold designed in exponential form displays a good effect. So, I set the threshold x_n between level n and level $n+1$ as:

$$x_n = 10^{\frac{-9+n}{2}}, n = 1, 2, ..., 9$$

2.2.4 Excellent User Experience

In the application, users could select the region in the analysis part and mark the start and end point in the navigation part by means of clicking the map. Besides, not only the click event could change the parameter showed in the map, but also the parameter in spin box will set automatically according to the click position. The pipeline of click event signal could discribed as pictured followed.

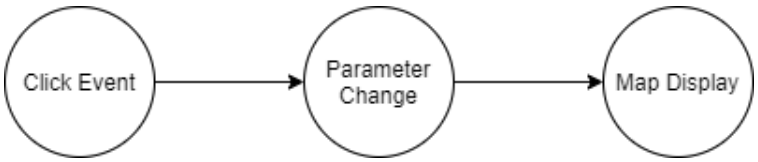

Fig.1.Click Event Signal Pipeline

Through the pipeline, both clicking the map and setting the parameter directly could change the map display. And using proper functions to manage the click event associating with the map condition, application could reach the effect showed in the demo.

2.2.5 Ubiquitous Wrong Input Detectors

Considering users may input illegal parameter, I set a lot of wrong input detectors in the application. For example, if the user want to set the start time later than the end time, the application will pop an error window and stop running. In addtion, if users want to plot an histogram with too many data point, the application will recommend user to change to use line chart.

3 Results

In this part, I will show you the efficiency comparsion between different optimized version, and the two different heat map display effect with different threshold.

3.1 Efficiency Expriments

As mentioned before, I designed three search versions during the optimizing process. To show the big effeciency differnt bewtween these versions, I plot a line chart of demand orders with the following parameters: *Customed; Areas:43, 44, 45, 46, 47; Start Day: 20161101; Start Hour: 0; End Day: 20161103; End Hour:2, Time Step: 0.5h* The parameters mean for each data series, the application should search in database 100 times, in other words, the application should search 500 times totally. The result is described in Table 1.

Table 1: Efficiency Comparision

Version	Time Cost
Version 1: Load into a table	16.78s
Version 2: Load into different tables according to days	6.06s
Version 3: Version 2 + area division	less than 0.6s

Obviously, the version 3 is much more efficent than the former 2 versions. The time cost is a tenth of the version 2, which will greatly enhence the user experience. Using version 3 to do some other expriments, we can feel its efficiency better.First, we test the application efficiency under one-area circumstance. The other expriments parameters and time cost are in the Table 2.

Table 2: Version 3 Performance under One-Area Circumstance

Area	Start Time	Start Hour	End Time	End Hour	Time Step	Maximum	Time cost
45	20161101	0	20161115	24	0.5	about 500	less than 0.6s
55	20161101	0	20161105	24	0.5	about 1300	1.2s
55	20161101	0	20161115	24	0.5	about 1300	3s
56	20161101	0	20161104	24	0.5	about 2000	1.3s
56	20161101	0	20161115	24	0.5	about 2000	4.5s

Given area 45, 46, 55, 56, 66 are the areas with the most number of demand orders, we choose these areas as the area parameter in the following multi-area expriments. The other expriments parameters and time cost under one-area circumstance are in the Table 3. The result shows that the time cost is not only proportional to the length of time but also proporional to the maximum number of demand order. In the worst condition, the one-area time cost is still under 5 seconds, and the multi-area time cost is about 10 seconds.

Table 3: Version 3 Performance under Multi-Area Circumstance

Start Time	Start Hour	End Time	End Hour	Time Step	Time cost
20161101	0	20161105	24	0.5	3.5s
20161101	0	20161105	24	1	1.9s
20161101	0	20161110	24	0.5	6.8s
20161101	0	20161110	24	1	3.5s
20161101	0	20161115	24	0.5	10.4s
20161101	0	20161115	24	1	5.4s

3.2 Heat Map Dispaly Effect

As mentioned before, I choose the posive proportional relation between the threshold and the level. However, the display effect is not good. The result is in Fig.2.

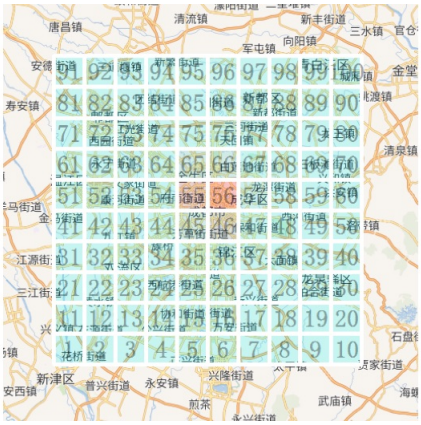

Fig.2.Proportional Relation

We could see that only 2 or 3 areas in the center of the city is red, while the other areas is almost blue. The relative amount of demand orders in different areas is not displayed in the heat map.

So, we choose a index relation between the threshold and the level. And we could see the distinct difference in different areas. The result is in Fig.3.

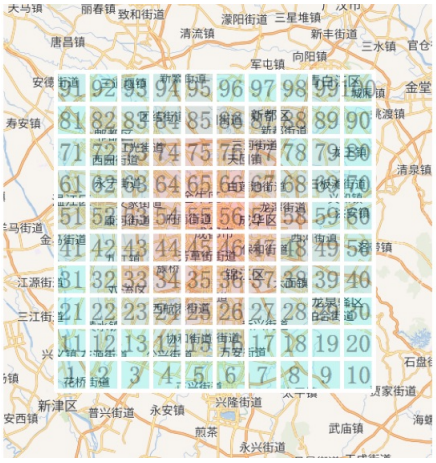

Fig.3.Index Relation

Using the index relation, we could see the the red fades from the middle to the periphery, which indicates the relative size of demand orders in different regions.

4 Discussions

In the application, users could choose the time range to load, and could cancel the loading process. In the analysis part, the application allow users to set a large number of parameters, providing users with plenty of freedom. The application is also robust, because it could detect many users' wrong input and warns users to correct. And it is convenient for users to input with mouse. Besides, the effiency of the application is far more increased by optimizing. Though the delay is obvious in the worst case, however, considering user usually won't use the application in the worst case, because user usually want to see the more detailed data in a smaller time range, the application performs well in most cases. The heat map display effect is also good, by using the index proportion into threshold settings.

However, there are still some problems to solve in the future. First problem is the long period of loading process, due to the preprocess. We may reduce the time cost by redesigning the data table. Second, though the time cost of the specific area is greatly reduced, the time cost of searching the total area is still high. By means of exploring other optimizing approach, it's possible to reduce the time cost. Last but not least, the navigation time estimate could be implemented if we mine the collected data.