Data Analysis and Visualization of POI Checking-In Data

Project for Principles and Practice of Problem Solving

Ning Li lining01@sjtu.edu.cn

1 Introduction

The aim of this project is to analyze the POI data provided by Gowalla and visualize it to demonstrate the underlying information. This project mainly focuses on the following information:

- The spatio-temporal information
  - The top 10 popular POIs visited by a user(or users) in a period of time.
  - The number of checking_ins of a POI(or POIs within a GPS range) in a period of time.
- The comparison information
  - The difference of visited POIs between two users over time.
  - The DAU of two POIs.
- Map-related analysis
  - The heatmap of POIs every day or in a period of time.
- Deeper mining of the data
  - The similarity between two users. Some comparison information will also be shown graphically here.

2 Implementation Details

2.1 Data organization and analysis

2.1.1 Reading Dataset

At first, The program(Hereinafter referred to as AVP(Analysis and Visualization Program)) provides a dialog for you to select file and fields. After you choose the file and fields that you are interested in, a thread will start to load and analysis the dataset. This prevent the program from being stuck.

AVP transforms data in .csv into five arrays named **user_id, location_id, time_id, la_id** and **lo_id** to store the five fields of each checking_in record. Considering the user_id has already been sorted, an integer array **user_pos** is used to store the position of different users in user_id. In order to find the checking_in records of a specific POI, AVP creates a two-dimensional integer array **location_count**.

2.1.2 Preliminary Data Analysis

When reading the dataset, AVP also creates a one-dimensional array **location_la_lo** to store the location_id and its latitude and longitude. With this information, AVP builds a **2-d tree**. The item's key is latitude and longitude and its value is location_id. The 2-d tree will help find POIs within a GPS range quickly.

As for the time data, Qt provides a method to transfer the string data into standard time data. However, since there are more than 1,500,000 checking_in records, The conversion progress will last more than 5 minutes. Considering that a lot of functions need the time data, such a long time is unacceptable. Therefore, for different functions, AVP provides different processing methods. If less time data is needed, AVP uses Qt method to transfer it. If so much time data is needed, AVP changes the year-month-day data into days to 2009-01-01 by manipulating the string data directly.

2.2 Spatio-temporal Information

2.2.1 The Top 10 Popular POIs

You can choose a specific user or a set of users. If you choose a set of users, you need to enter the user_ids you are interested in with the given format(For example:1,6-10,88,92, 120-167). AVP will transfer the input data into user_ids. Besides, you can choose the time range you are interested in. With these parameters, AVP will count the checking_ins of the POIs the users have visited. AVP will compare the time range by comparing the time string directly. In the end, a bar chart will show the top 10 popular POIs and the number of their checking_ins.

2.2.2 The Number Of Checking_ins

You can choose a specific POI or POIs within a GPS range.

If you choose a specific POI, AVP will display a line chart. It will show the number of checking_ins each day. You can learn about the checking_in trend of the POI. AVP has implement the magnification function of the line chart(all of the bar charts and line charts in AVP can be enlarged unless specifically stated). By selecting a range horizontally on the line chart with the mouse, you can enlarge part of the line chart.

If you choose POIs within a GPS range, you can enter the latitude range and the longitude range. Also, you can choose the time range you are interested in. With these parameters, AVP will show a bar chart whose x axis is location_id and y axis is the number of checking_ins. Sometimes the are so many POIs that it will be very narrow if all POIs are shown in the window at the same time. To solve this problem. AVP overloads the QChartView class. It only shows 10 POIs at the same time. you can click and drag the chart to the left, then other POIs will appear.

2.3 Comparison Information

2.3.1 The Difference Of Visited POIs

Choose the user_id of user A and user B, then AVP will show a bar chart. The chart shows the number of POIs visited by user A, suer B in each month. It also shows the number of POIs visited by user A and B in each month. The reason for showing in months is that there is enough checking_in data in a month.

2.3.2 The DAU Of Two POIs.

Choose the location_id of POI A and POI B and the display form, the AVP will show a line chart or a bar chart.

If you choose the line chart, it will show the DAU of POI A and POI B in each day.

Not only can you compare the DAU of two POIs in each day, but also you can learn about the DAU changing trend over time. If you choose the bar chart, you can learn about the specific value of DAU. You can choose the time range you are interested in, but I recommend you to choose time range in ten days.

What should be noticed is that the DAU of a POI is different from the number of checking_ins in a specific day, because a user can visit a POI for many times in a day. AVP creates a class named DAU to manipulate the data. According to the given POIs, AVP scans the checking_in records, divide them into different days, and use std::set to remove the visited records of same users in a day. The divide progress will transfer the time string into QDateTime. If the two POIs have some many checking_in records, it will cost a few seconds.

2.4 Heatmap

You can choose the heatmap of a single day or a period of time.

If you choose a single day, you can see the heatmap in the first day of all records. If you click the heatmap, you can see the heatmap of the next day that has checking_in records. By doing this, you can learn about the changing of the distributions of the popular POIs. If you choose a period of time, you can see the heatmap in the time range you are interested in.

The x axis of the heatmap is longitude and the y axis is latitude. The colors of the heatmap are from blue-green-yellow-red. AVP counts the number of checking_ins of every POI in a day(or a period of time). The POI with the most checking_ins weighs 1. Other POI's weights equal to their checking_ins÷(the largest checking_ins). Each POI is a circle. Its center point weight is the POI's weight and its edge weight is 0. Different circles are superimposed on each other, forming a weight map. The weights correspond to the colors. Then a heatmap comes into being.

2.5 Similarity

Table 1: Parameters and computing progress

Parameter	Explanation
H	The number of the common active hours between user A and user B
D	The number of the common active days between user A and user B
AD_1	The most active day of user A
AD_2	The second most active day of user A
BD_1	The most active day of user B
BD_2	The second most active day of user A
NA	The number of POIs visited by A
NB	The number of POIs visited by B
N	The number of POIs visited by A and B
P_1	$P_1 = H \div 6$
P_2	$P_2 = D \div 2$ if D=1 or D=0 P_2=1 if D=2 and AD_1=AD_2 and BD_1=BD_2 P_2=0.75 if D=2 and AD_1=BD_2 and AD_2=BD_1
P_3	$P_3 = (N \div NA + N \div NB) \div 2$

Choose the user_id of user A and B, then you can get an analysis of how similar they are. First, According to the number of checking_ins in each hour, AVP gets the top six active hours of each user. According to the number of checking_ins in each day of week, AVP gets the top two active days of each user. According to the records of user A and B, AVP gets the number of POIs visited by A, B, A and B. P_1, P_2, P_3 are 3 parameters of the similarity. Parameters and computing progress are shown in Table 1.

The weight of P_1 and P_2 is 0.9, The weight of P_1 is $24 \div (24+7) * 0.9 = 0.7$. The weight of P_2 is $7 \div (24+7) * 0.9 = 0.2$. the weight of P_3 is 0.1. Therefore, the similarity $S = P_1 \times 0.7 + P_2 \times 0.2 + P_3 \times 0.1$.

Some aspects of the similarity can also be shown graphically. You can choose "similarity of day" or "similarity of week". The former will display the number of checking_ins of A and B in each hour of day with bar chart. The later will display the number of checking_ins of A and B in each day of week.

3 Results

There are so many functions in AVP, so I choose some of them to demonstrate.

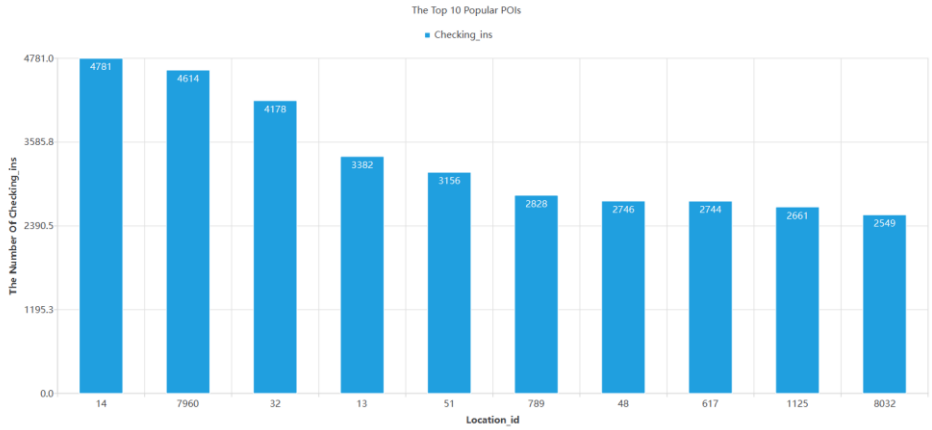

Figure 1:The top 10 popular POIs visit by all users.

From Figure 1 we can know that the numbers of checking_ins of the top 10 popular POIs are all larger than 2500. But the average checking_ins of all POIs is about only 30. Therefore, we can deduce that a few POIs account for most of the checking_ins.

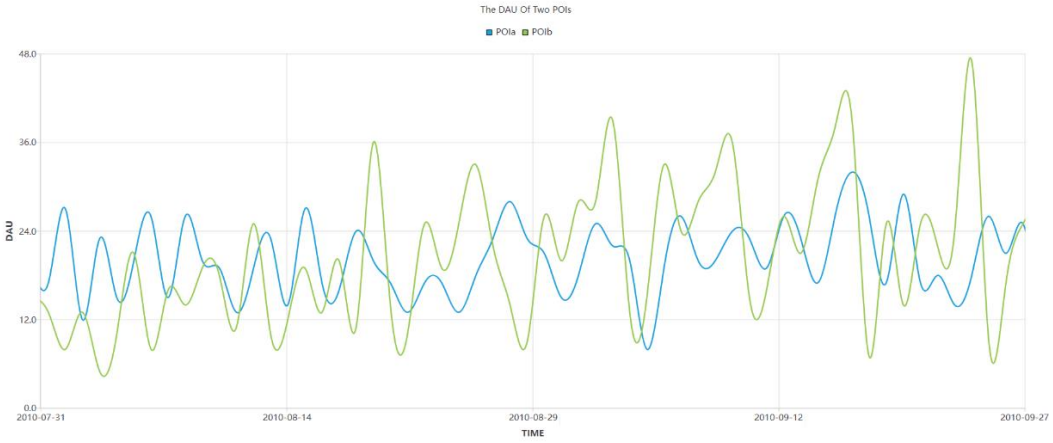

Figure 2: The DAU of POI 14 and POI 7960.

According to Figure 1, we choose the top 2 POIs to compare. We enlarge part of

the line chart, as Figure 2 shows. We can see the changing trend of DAU between two POIs. What should be noticed is that compared to POI 14, the DAU of POI 7960 is more cyclical. But the DAU of POI 14 is more stable.

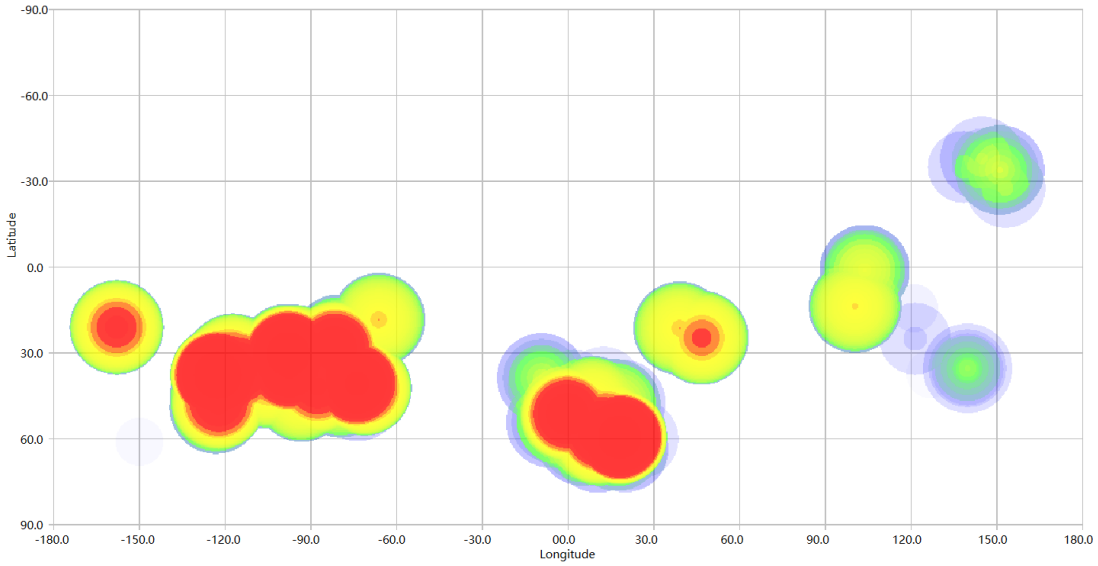

Figure 3: The heatmap from 2009-1-1 to 2010-12-31

The heatmap including the checking_in records in all time range is shown in Figure 3. We can know that the hot spots are located in a few regions. Most of the regions on the earth have few checking_in records. But it has to be acknowledged that 71% of the earth is covered by ocean.

4 Discussions

4.1 Characteristic

1. The application implements several interactions between the chart and mouse. For example, dragging the chart to display different POIs(**2.2.2**)", click the chart to display heatmap in the next day (**2.4**).
2. There are various display forms, such as bar chart, line chart, heatmap and so on.
3. The application has good robustness. The user_id, location_id and time are limited. When using this application, it is almost impossible to exit abnormally.

4.2 Evaluation

Since that the application implements a specific method to manipulate the time data, the computing time has been greatly reduced. We test the top 3 time-consuming tasks. The results are shown in Table 2.

Table 2 The elapsed time of tasks

Task	The number of tests	Average Time(ms)
Read Dataset	5	7836.2
Compare POI 14 and POI 7960	5	2325
Draw Heatmap at all time	5	2049.4