Data Analysis and Visualization of POI Checking-In Data

Project for Principles and Practice of Problem Solving

1 Introduction

Point-of-interest (POI) data has been widely used in many applications including location-based social networks, navigation software, recommender systems, etc. In this project, you will be asked to analyze the POI data provided by Gowalla and visualize it to demonstrate the underlying information, like the tendency of the flow of people for a given place. We hope your work can help mine valuable information, which is useful in aforementioned applications.

2 Dataset Description

In this project, the provided dataset is extracted from the location-based social networking website Gowalla. The dataset is given in the format of a CSV file. Each line in the CSV file corresponds to one POI checking-in record, and there are altogether 1,502,536 records. Each record contains five fields as listed in Table 1.

Table 1. Fields of checking-in record.

Field	Data Type	Description	Example
user_id	int	user ID	0
location_id	int	location ID	0
time	string	timestamp of POI checking-in	2010-05-22T02:49:04Z
latitude	float	latitude of POI	30.248923845
longitude	float	longitude of POI	-97.74962604049999

In the example above, user with ID 0 visited the POI with ID 0 located at (30.248923845, -97.74962604049999) at 02:49:04 on May 22, 2010.

3 Tasks

3.1 Mandatory Part

Design a graphic user interface, which

1) allows the user to **load the dataset**. Implement a **filter** to import the selected fields of user's interest. A progress bar is required to show the loading progress.
2) can display the spatio-temporal information. For example, the top 10 popular POIs visited by a specific user (or a set of users) over time, the number of checking-ins of a specific POI (or POIs within a GPS range) over time. Display forms include but not limited to the line chart, histogram, pie chart, etc. The user should be able to **tune the parameters** including starting time, ending time, user IDs, POI IDs graphically. **Interpolation** is required to smooth the line when only a few data points are available.
3) can show the comparison information. For example, the difference of visited POIs between two users over time, the daily active users (DAU) of two POIs.

Note: The checking-in records for some users and POIs might be too sparse to support meaningful data analysis and visualization. You may focus on those with considerable number of records, and we will also check your submission with them.

Hint: Codes need to be implemented via multi-thread to prevent your program from being stuck when you are computing while displaying. KD tree algorithm may be helpful when searching POIs that locate in the GPS range chosen by user.

3.2 Elective Tasks

Implement additional analysis by yourself. Use your knowledge and freely exert your creativity, and we won't limit views from which data is analyzed and displayed. Please note that one can implement up to two functions in elective tasks. Functions that are more than two will not be counted in the total score. Here are some recommendations:

– Map-related analysis.
  - Show the trajectory of a specific user;
  - Show the heatmap of popular POIs, i.e., plot the thermal diagram in a period of time;

  Hint: We recommend openstreetmap library.

– Prediction-related analysis.
  - Given a POI, how many users will visit here in next hour (day, week, month)?
  - Given a user, which POIs he/she may visit next with high probability?

– Deeper mining of the data.
  - Define your similarity metric between users (or POIs). Now with the checking-in data, how similar are two users? I.e., if two users visited many POIs in close periods, then they are quite in common.

4 Submission Requirements

The project will graded on the basis of your **project report**, **video demo** and the **source codes**. You need to zip and upload these files to the course website. It is due by **11:59pm on January 2th, 2022**. **No late submission will be accepted.**

4.1 Implementation Requirements

The graphic user interface should be implemented with Qt in C++.

4.2 Report Requirements

1. No more than 5 pages. No formatting requirements.
2. Should at least contain
   a) **Introduction.** Describe your main ideas and what you do in this project.
   b) **Implementation Details.**
   c) **Results.**
   d) **Discussions.** You could discuss the performance of your application or discuss the interesting results you have revealed from the data.
3. Report is in English only.

4.3 Demo Requirements

1. Make a video to present your work by screen recording or camera shooting.
2. No more than 5 minutes.
3. Presentation is in English only.

Examples of screen recording:

– YouTube Demo
– Bilibili Demo

Examples of camera shooting:

– Presentation Demo

4.4 Source Code Requirements

1. Add a README file, which includes instructions to run your codes.
2. Make sure it runs smoothly on your own computer.