1 Proposed project:

Road Trip Planning Assist

2 Motivation:

Recently I've been planning a family road trip for my family visiting California. While planning the tourist attraction points and accommodations, I've noticed that even with using Google Maps the task has been quite tedious. This task could and should be able to be much easier for everyone.

3 Introduction:

When one is on a vacation travel, a business trip, or even looking for a new place to stay, one would have to look up the area and find a location that is best for one's purpose. For example, John is on a vacation travel and would like to visit the San Francisco bay area without missing any of the fun stuffs within a limited of time. While John checks with Google Maps online, he soon finds that the physical distance of two places doesn't really matter that much, what he cares is the traveling(driving) time. The traveling time could be inconsistent with the physical distance due to many factors, such as topography, traffic jams, road designs, time of day, weather, etc. That is, for a traveler/commuter, the distance of two locations is warped by many of these factors.

Another example case would be Jane finding a housing location. Compared with John's short term trip, Jane's case would be a long term relocation. The basic idea would be the same; however, the points of interest would be different. For example, Jane would be more concerned about bank ATM locations instead of where gas stations are located.

Fortunately, powerful online services such as Google Maps has done most of the hard work for us. Navigation, route planning, time estimation, traffic, etc. are all included within its service. However, in order for someone to plan for road trips, business trips, and/or housing location searching without testing out all the combinations to find an optimal choice, we could make use of Google Maps API to gather traffic and travel data.

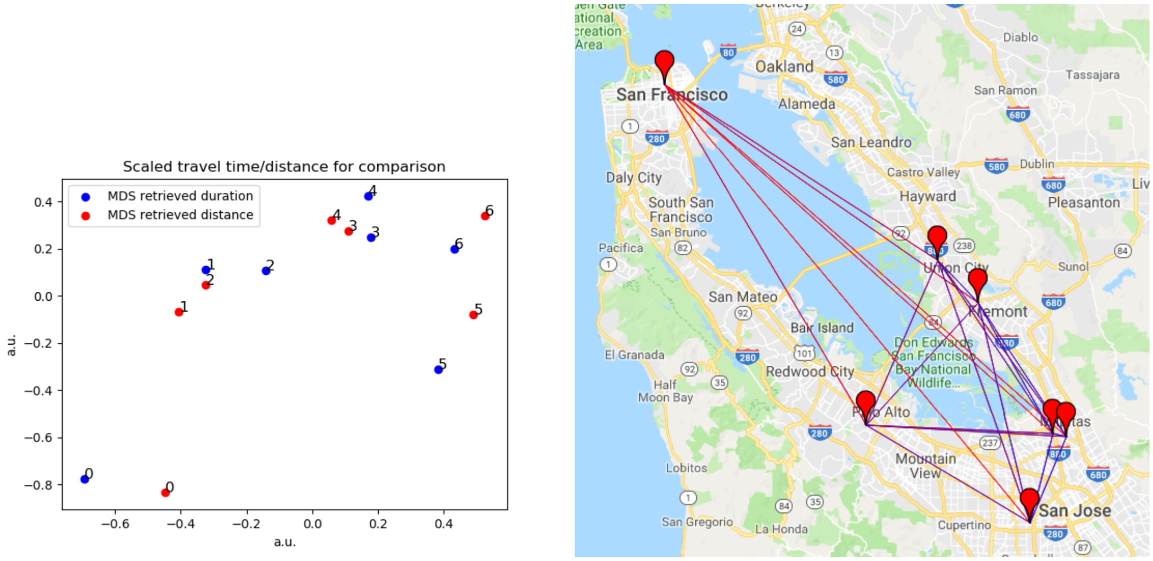
4 Goals:

Come up with a solution to help users simplify the task of finding suitable accommodation locations. Multiple locations and their distances/times should be able to be easily summarized and visualized for the user.

5 Work sketch as of May 5:

This project idea came from my complaints last week of travel planning. The type of data and tools were all introduced to me today (May 5), so I'm not familiar with any of these; however, they are quite fun!

- Got Google Maps API to work, the queries from this API are used as data for my toy example. Arbitrary locations are used in this example.
- Applied MDS and used its results to make a quick comparison of the physical distance v.s. travel duration. This plot shows how travel time could be different to the physical distance depending the time of day or other factors. (Of course, true distance is still highly correlated the travel time.)
- Arbitrary locations are marked on a map with color coded connecting lines. The redder(bluer) the line, the longer(shorter) time it takes to travel between these two points.
- Other analyses can be done after pulling out more data.



(a) MDS plot for physical distance v.s. travel duration.

(b) The redder(bluer) the line, the longer(shorter) time it takes to travel between these two points.

Figure 5.1: Toy example comparing physical distance with travel duration using arbitrary locations. Demo code: `https://bit.ly/30i8wie`

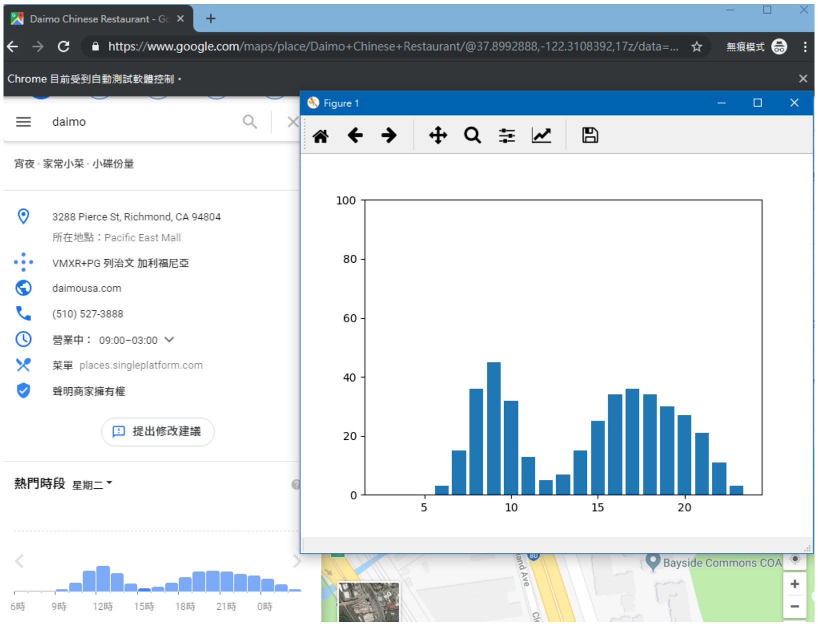
6 Defining our problem:

Let's say we have 3 points of interest, A, B, and C in a certain area we are planning to visit. All of these POIs are reachable by driving/public transportation. We then have 2 tasks to solve: 1. Determine the staying location 2. Our schedule for the trips Both tasks depend on the traveling times among the locations, which the traveling time itself is a function of the day of the week and time of day. The schedule would also depend on more information, such as open/close time, popularity time (we would prefer to avoid the crowd), and normal time duration of the stay.

If we have 3 locations O_1, O_2, and O_3 which we prefer, we can evaluate which location has a lower cost in traveling time with respect to POIs A, B, and C. If no preferred origins are given, we should be able to draw out a contour plot or an area on the map to show a suggested area that gives a relatively low cost in travel time with respect to the POIs we've given.

7 Data source:

- Google Maps API
- Scraping Google Maps for information not provided by Google Maps API (e.g. popularity info, traffic data for time of day and day of the week)



Daimo Chinese Restaurant - Go

https://www.google.com/maps/place/Daimo+Chinese+Restaurant/@37.8992888,-122.3108392,17z/data=...

Chrome 目前受到自動測試軟體控制。

daimo

宵夜 · 家常小菜 · 小碟份量

3288 Pierce St, Richmond, CA 94804

所在地點：Pacific East Mall

VMXR+PG 列治文 加利福尼亞

daimousa.com

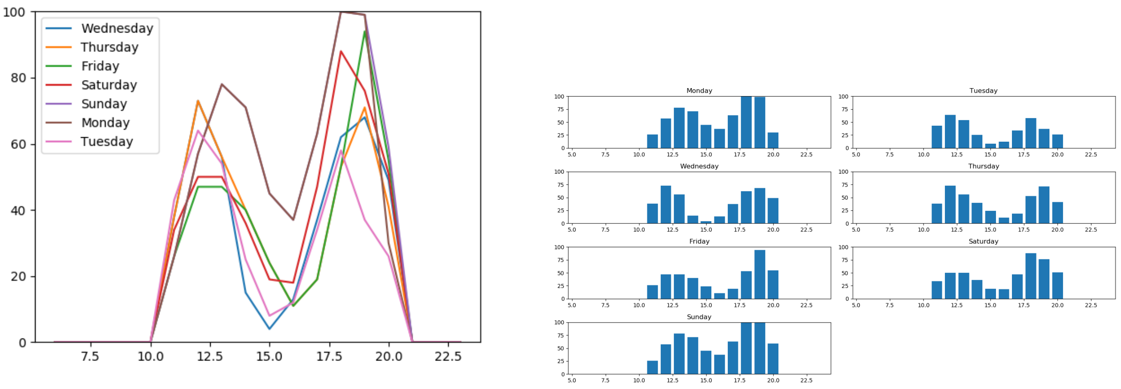
(510) 527-3888

營業中： 09:00–03:00

菜單 places.singleplatform.com

聲明商家擁有權

Figure 7.1: Toy example of scraping popularity data from Google Maps. Demo code: `https://github.com/abcshia/test/blob/master/scrape_gmaps_pop_toy_example.py`.



(a) Location popularity data over a week.

(b) Bar plot of location popularity data over a week.

Figure 7.2: Toy example comparing location popularity data. Demo code: `https://bit.ly/2VuQk16`

- Scrape/Query(osmapi) OpenStreetMap and Nominatim for general data (reducing the reliance on Google Maps)

8 Deliverables:

1. A summarized travel time table for different origin/destination combinations, including time of stay, time of the day, and day of the week.
2. If origins are not given, a suggested area that fits the input condition would be given.
3. A visualized map should be plotted to show the results easily.

9 Possible extension work:

Add in other cost factors instead of travel time:

- Include gas stations info: Scrape GasBuddy, use mygasfeed api
  (http://www.mygasfeed.com/keys/intro.)
- Include bank ATM locations for relocation uses.
- Include altitude info for trip planning.
- Include hotel information:
  E.g. KAYAK Search-Widget API

- Include other map service info:
  E.g. Zillow API, yelp API, etc

Examine the data of different combinations of POIs, collect and analyze the distributions of suggested paths (roads, streets) that are given.