ECE 232E Project 4 Report

Ken Gu, 904-836-308

Yingbo (Max) Wang, 604-593-537

Ziying Feng, 404-395-859

Siqi Huang, 504-490-530

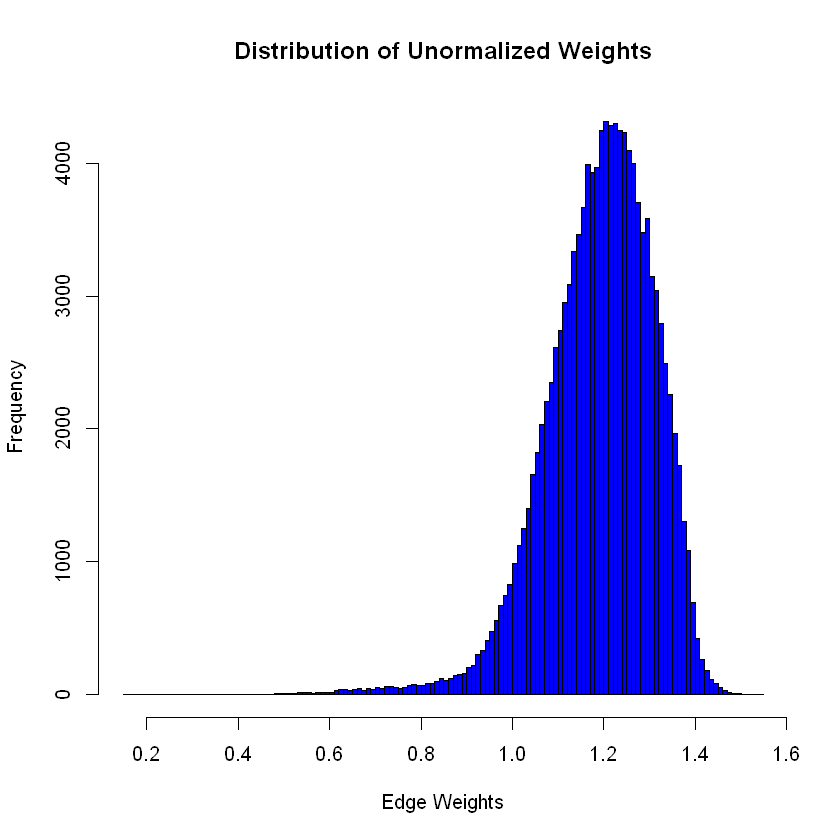
Question 1

As the question mentions correlation between two time series values, we can view this as the correlation between two random variables. Using knowledge from prior probability courses we note the formula for the Pearson correlation coefficient.

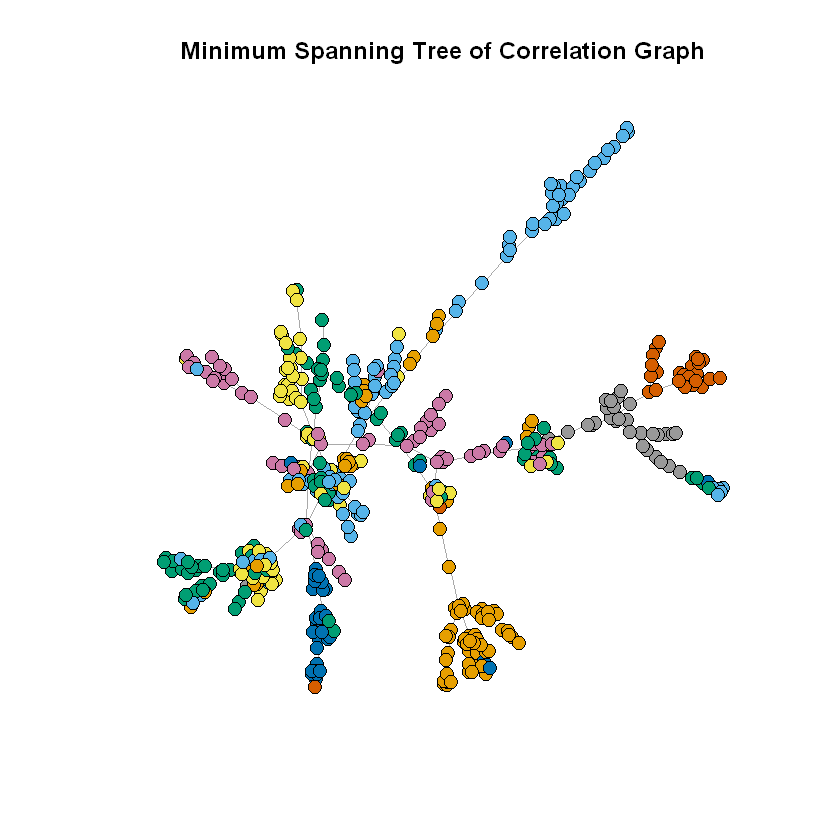
This can also be written as

This is the exact same expression as the one in the question with and operator representing the expectation ( ) operator. Thus, same with the Pearson correlation coefficient the upper and lower bound for is 1 and -1 respectively.

An argument for using log-normalized return is since and if we assume that stock prices (are log normally distributed which historical stock prices often show to be then is normally distributed which makes it easier to model.



Question 2

Question 3

We can see there are some main edges that essentially act as main branches (if we go by the vine analogy) that connect between different clusters (clusters being stocks of a specific sector). Moreover, at these clusters there are smaller edges that connect stocks of the same cluster. This is like a cluster of grapes at the end of the vine. This shows that nodes of the same cluster are more likely to be connected to other nodes of the same cluster and are relatively close in distance. Essentially this means edges between nodes of the same cluster have smaller weights which makes sense since the weight has the term . In other words, nodes in the same cluster or close to each other in distance means they have higher correlation with each other.

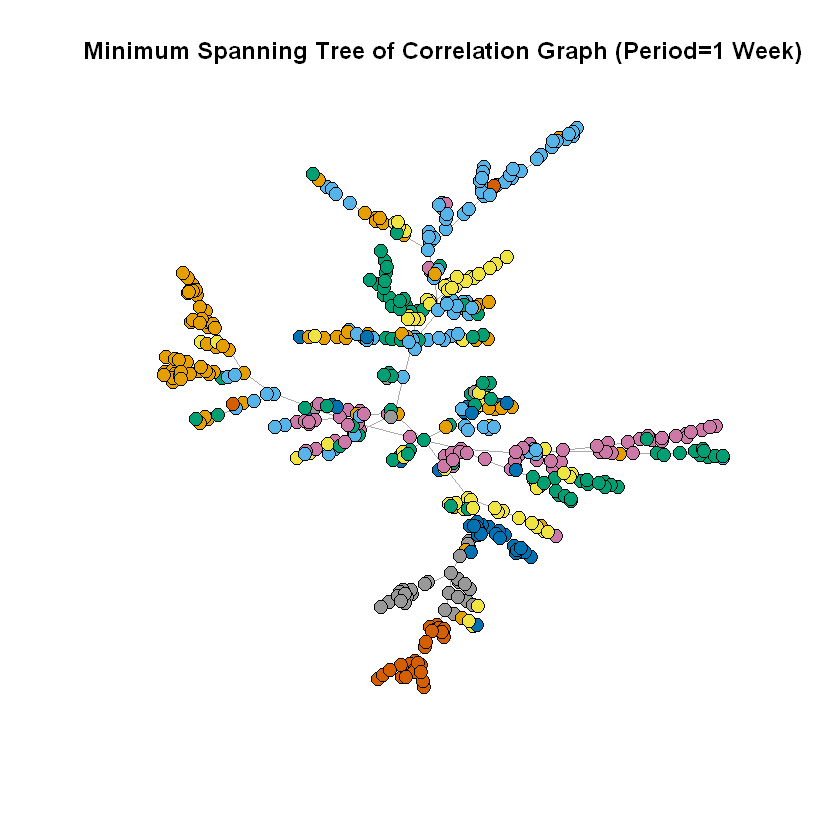
Question 4

when

when

The difference in the alpha values reveals very different ways in evaluating the performance of the methods. For the first alpha definition we are evaluating within similar predicted companies to a given company, how many other companies are from the same sector. It is a measure of a node belonging to a sector based on the sectors of its neighbors. This is good for evaluating a prediction for market sector of an unknown stock based on what are stocks are like it (node neighbors) from the MST method. For the second alpha definition we are essentially calculating . This can be interpreted as a measure of a company belonging to a sector solely based on the number of companies in that sector divided by all the companies. This makes no use of graph information which is a reason why the alpha value is so low.

Question 5



The MST from the correlation graph based on weekly data is very similar to the MST on daily data in that nodes in the same cluster are close to each other and are off a main branch like a vine. However, a key difference I’ve noticed is that the MST from the correlation graph based on weekly data is better able to separate the cluster of stocks near the left center of the MST on daily data into individual branches along the main vine branch.

Question 6.

As explained in the specification, we read the San Francisco Monthly Aggregated Traffic data file line by line and select entries that has “month” attribute equals 12. One edge is created between each pair of source and destination node IDs with weight equals the average traveling time between the two nodes.

This graph has 2710 vertices and 321713 edges excluding the extra nodes in the Geolocation metadata file. Then we find the Giant Connected Component of this graph; it has 1898 vertices and 321703 edges. Thus, about 70% of the nodes form the biggest cluster with about 99.997% of the edges in the original graph.

Question 7.

The Minimum Spanning Tree has 1898 vertices and 1897 edges. 15 edges in the MST are randomly sampled and presented in the following table. We include the edge ID as a tuple of (source node ID, target node ID), the average traveling time between the two nodes, and the address and coordinate of both the source location and target location.

The results make sense as they correspond to the travel time given on Google Maps. Locations pairs that have higher travel times are either geographically farther away or with frequent heavy traffic in between. For example, it’s a relatively short drive from “3600 Parsons Court, Castro Valley” to “13700 Westboro Drive, Alum Rock, San Jose” of 39.2 miles – Google Maps estimates it to take 55 mins with medium traffic and our dataset shows 67.05 mins, which is very close. On the other hand, a drive from “200 North 5th Street, Central San Jose, San Jose” to “5700 Wildwood Drive, Marysville” is 154 miles and takes much longer with about 3 hour and 16 mins estimated by Google Maps. That’s close to the average traveling time in our dataset of about 225.455 minutes.

Table. Street Addresses of Two Endpoints on a few Edges

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Edge | Average Traveling Time | From Address | From Coordinate | To Address | To Coordinate |
| (833, 1433) | 244.71 | 200 El Cajon Avenue, Covell Park, Davis | (-121.752, 38.565) | 8100 Dunes Court, Sacramento | (-121.402, 38.459) |
| (165, 1694) | 163.215 | 2200 Franklin Street, Pacific Heights, San Francisco | (-122.424, 37.794) | 700 Folsom Street, South of Market, San Francisco | (-122.402, 37.782) |
| (850, 1166) | 82.685 | 1300 Greenborough Drive, Cirby Ranch, Roseville | (-121.275, 38.723) | 600 Cinnabar Court, Diamond Oaks, Roseville | (-121.28, 38.764) |
| (927, 1375) | 135.85 | 800 Aspen Way, Palo Verde, Palo Alto | (-122.109, 37.426) | 700 Golden Avenue, Lodi | (-121.259, 38.146) |
| (978, 1383) | 226.095 | 7600 Wilbur Way, Sacramento | (-121.398, 38.477) | 100 Tomlinson Drive, Folsom | (-121.176, 38.702) |
| (424, 1703) | 149.23 | 5100 61st Street, Southeastern Sacramento, Sacramento | (-121.437, 38.529) | 2600 21st Street, Mission District, San Francisco | (-122.406, 37.759) |
| (506, 923) | 175.005 | 2700 Marty Way, Southwestern Sacramento, Sacramento | (-121.494, 38.557) | 1300 North Carpenter Road, Modesto | (-121.033, 37.657) |
| (545, 807) | 225.455 | 200 North 5th Street, Central San Jose, San Jose | (-121.889, 37.342) | 5700 Wildwood Drive, Marysville | (-121.55, 39.124) |
| (423, 1105) | 152.64 | 200 Locust Avenue, San Rafael | (-122.508, 37.982) | Unnamed Road, Castro Valley | (-121.987, 37.621) |
| (364, 1171) | 114.755 | 5300 Laguna Park Drive, Elk Grove | (-121.433, 38.43) | 3100 Forest Lake Road, Pebble Beach, Del Monte Forest | (-121.945, 36.588) |
| (1266, 1269) | 243.805 | Spring Creek Trail, Santa Rosa | (-122.666, 38.451) | 500 Yosemite Drive, Milpitas | (-121.899, 37.421) |
| (1658, 1820) | 154.725 | 6800 Fairmount Avenue, El Cerrito | (-122.298, 37.902) | 2900 Glacier Street, Bohemian Village, Sacramento | (-121.408, 38.62) |
| (601, 1274) | 67.05 | 3600 Parsons Court, Castro Valley | (-122.084, 37.706) | 13700 Westboro Drive, Alum Rock, San Jose | (-121.827, 37.359) |

Question 8.

We first find all triangles in the Giant Connected Component graph by getting cliques with size 3. Then 1000 triangles are randomly selected and weights of its three edges are tested with the triangle inequality. We find that the percentage of those triangles satisfying the triangle inequality equals about 92.60%.

The result makes sense as most of the times going from place A to place B directly is faster than adding an intermediate stop (going from place A to place C and then from place C to place B). The few exemptions can be caused by traffic conditions: the direct route between the two locations may usually have heavy traffic, so going a longer detour with less traffic could actually save time.

Question 9.

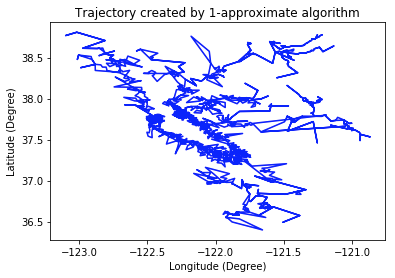
To calculate the cost of the 1-Approximate TSP algorithm, we first implement the Tree Algorithm to find the optimal path that Santa travels: it equals an Eulerian circuit of the multigraph of the graph’s MST, and then only keeps its unique nodes (with a shortest path connecting the pairs without an edge connecting the two nodes). The Eulerian path can be calculated by first finding a random cycle on the multigraph from a starting node and then repeatedly adding cyclic paths starting from nodes already in the path, until no node in the path is incident to a remaining edge.

The generated Eulerian path travels 3795 steps and is presented as a list of node IDs EULERIAN\_PATH.py. The calculated Santa’s tour path travels 2250 steps and the list of node IDs is presented in TOUR\_PATH.py. The result makes sense according to the following visualization in the next question, because the traveler thoroughly visits every location in the Northern California area while adjacent locations are reasonably close to each other. A list of coordinates in this tour is shown in tour\_path\_coordinates.csv.

Then we sum up the weights of all traversed edges to get the cost of this tour to be 504399.755. Also, the cost of the optimal TSP path is lower bounded by the cost of MST that equals 289315.675 (the summed weights of all edges in the graph’s MST). Thus, in this problem, the upper bound on the empirical performance of the approximate algorithm equals:

The result makes sense because it’s less than two times the cost of MST.

Question 10

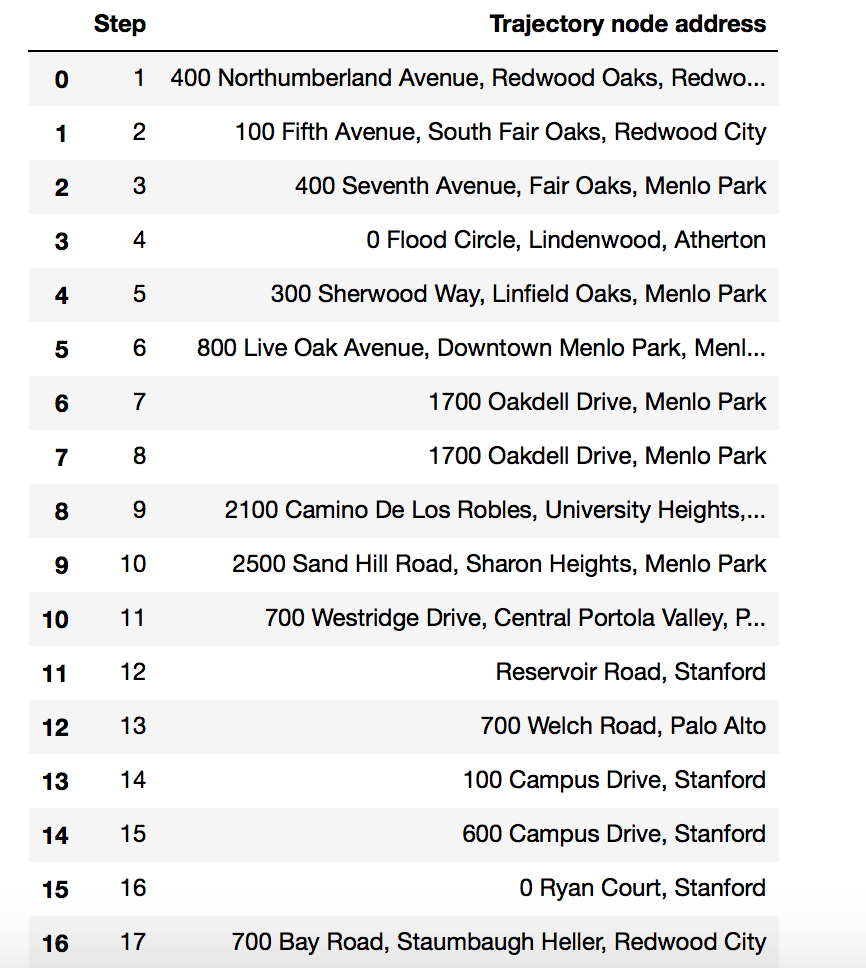


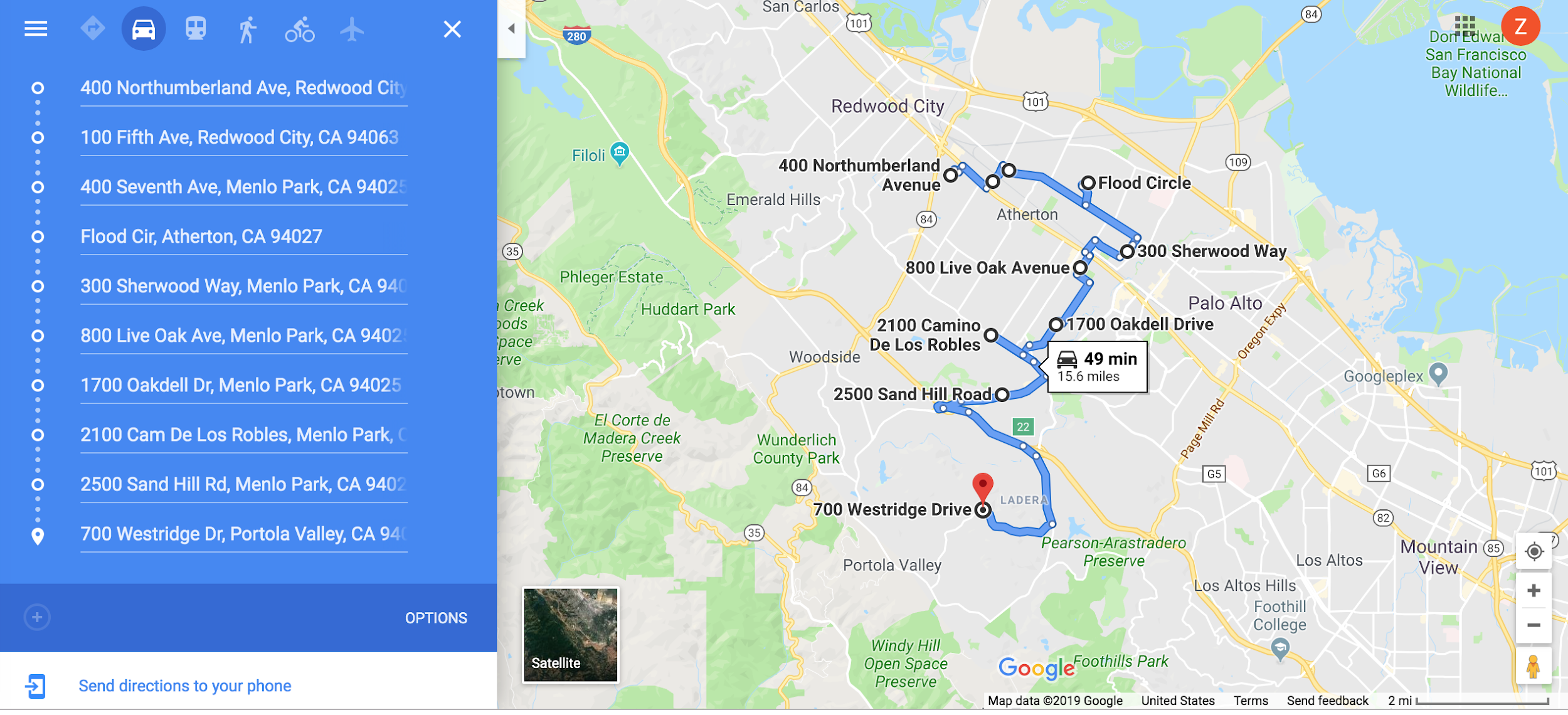
|  |  |
| --- | --- |
| 1-Approximate Algorithm | Minimum Spanning Tree |
|  |  |

The figure above compare the trajectory plot of Santa has to travel generated by the 1-approximate algorithm and the minimum spanning tree in igraph package. They look very similar, which means that the 1-approximate algorithm works very well.

The minimum spanning tree generated by the igraph package uses the Prim’s algorithm, which is a kind of greedy algorithm. That is “We start from one vertex and keep adding edges with the lowest weight until we reach our goal”.

The dataframe below is the initial part of the address sequence produced by 1-approximate algorithm. And the following is the first 10 street addresses in the sequence showed onto the real map. The results make sense, because the adjacent nodes in the sequence are close to each other.





Question 11

|  |  |
| --- | --- |
| Road mesh | Zoom-in road mesh |
|  |  |

The figures above are the road mesh created by Delaunay triangulation. After Delaunay triangulation all the nodes are connected, and the Delaunay algorithm maximize the minimum angles of all the triangles. For each triangle its circumcircle contains no other nodes, and for each edge a circle exists through its endpoints contains no other nodes. The Euclidean minimum spanning tree (the Euclidean minimum spanning tree is the minimum spanning tree based on the Euclidean distance of the edge in space instead of the weight of the edge) is a subset of the Delaunay triangulation of the same set of nodes.

Question 12

, is cars/(road.hour), is cars/mile, is mile/hour

Safety distance:

Distance between two cars:

Density:

Flow:

Each road has 2 lanes, the flow is:

|  |
| --- |
|  |

To calculate the flow of each road the edge of the graph, the edge distance is calculated by the Euclidean distance between the coordinates and then convert the degree to mile, using average 69 mile per degree. The time is the mean travel time of the edge. So Then use the equation derived above to calculate the flow of each edge. The average flow of the graph is about **2996 cars/(road.hour).** Below is part of the flow values of the edges:

|  |
| --- |
| 3281 2706 3157 3043 3312 2969 3040 3327 2858 3243 3266 2494  3024 3042 3107 3030 3202 3229 2846 3160 2447 3031 3234 2765  3277 3163 2646 3169 3270 3262 2951 3069 3193 2951 2887 2920  3137 3179 3234 3074 3148 3147 2692 3210 3317 2950 3169 3296  2949 2913 3163 2717 3254 3302 2499 3172 2889 2744 3082 3231  3236 3005 2771 3066 2757 2806 3141 2980 3316 3238 3151 3183  3286 2816 3054 3294 3171 3242 2753 2671 3240 3081 2827 2554  2581 3299 3226 2922 3098 3131 3119 2497 2957 3256 3156 2822  2677 3272 2878 2674 2785 3309 3126 3031 3099 3137 3160 3018  2902 2969 3286 3159 3148 2941 2700 3165 3096 2827 2500 3276  2914 3117 2927 3206 3078 3169 3208 2791 3017 3186 3230 3301  2167 2892 2788 3184 3158 2521 2918 3140 2985 3196 3254 3223  2860 2600 2749 3148 2199 2617 2418 3189 3055 2924 2946 3195  3330 2886 3016 2754 3321 3127 3067 3052 3107 3107 3085 3034  2715 2754 2715 2757 2851 3112 2674 2699 2641 3088 2770 2658  3242 3024 3150 3406 3189 3101 3167 2990 3313 3075 2707 2621  2440 3263 2981 3060 2751 3100 3144 2898 3046 3100 3191 2869  3036 2919 3250 3046 3257 3281 2974 2364 2327 2750 3152 3155  2772 3023 2936 2660 3093 3310 2733 2765 3095 3076 3145 2881  … ... |

Question 13

Use the maxflow algorithm Ford-Fulkerson algorithm to calculate the maxflow of the graph based on the flow of each edge calculated in Question 12. The maxflow value is **470908 cars/(road.hour)**. It means that in the ideal case (no traffic jam), the maximum total number of cars that can commute on the road travelling from “100 Campus Drive, Stanford” to “700 Meder Street, Santa Cruz” is 470908 per hour.

The number of edge-disjoint paths equals the number of edges that have to be removed in order to disconnect the two vertices. The number of edge-disjoint paths between the source and the destination is **6**.

**Edge-disjoint paths are the maximum number of paths that don’t have common edges. The number of edge-disjoint paths problem is one of the variants of the maxflow problem.** In the edge-disjoint paths problem, each edge has a capacity of 1. And the maxflow between the source and the target nodes in this graph is k if and only if the number of edge-disjoint paths is k.

The results make sense, because from the zoom-in road mesh of source and the destination we can see that there are 6 neighbor nodes directly connected to Stanford node, but there are only 5 neighbor nodes directly connected to UCSC node. So the maximum number of paths that don’t have common edges should be a number that is <= 6.

|  |  |
| --- | --- |
| Stanford neighbourhood (-122.18, 37.43)  ID in Geo data = 2607 | UCSC neighbourhood (-122.06, 36.97)  ID in Geo data = 1968 |
|  |  |

Question 14

|  |  |
| --- | --- |
| Pruned graph | Original Delaunay graph |
|  |  |

Delaunay triangulation might generate some edges that are not in the graph. To prune the graph:

(1) Check all the edges of all the triangles generated by Delaunay to see if the edge is in the edgelist of the graph;

(2) If the edge is in the edgelist of the graph, then check if the edge mean\_travel\_time is lower than the set threshold 600;

(3) Remove the edge if it is not in the edgelist of the graph, or if the edge mean\_travel\_time is lower than the set threshold.

For the pruned graph now, all the edges are “real road” that is they are in the edgelist of the graph, also all of their mean\_travel\_time is within 600 seconds (10 min).

As for the bridges, from the pruned graph, we can see that all the bridges except for the “San Mateo Bridge” are treated as fake bridge and removed, because it is the longest bridge of the five bridges listed in the question leading to longer mean travel time.

Question 15

We repeat Question 13 with the pruned graph produced in Question 14, that is (1) calculate the flow of each edge of the pruned graph, (2) get the maxflow and the number of edge-disjoint paths between Stanford to UCSC.

The maxflow value of the pruned graph is **21363 cars/(road.hour)**. It means that in the ideal case (no traffic jam), the maximum total number of cars that can commute on the road travelling from “100 Campus Drive, Stanford” to “700 Meder Street, Santa Cruz” is 21363 per hour. And the number of edge-disjoint path remains **6**.

The maxflow value of the pruned graph is lower because the threshold is relatively low, leading to a lot of road are treated as “fake”, according to the Ford-Fulkerson algorithm for the calculation of the maxflow, since there are less road that could be used to travel from Stanford to UCSC, the number of cars that can commute between these two nodes will decrease.

From the zoom-in road mesh of source and the destination we can see that there are 6 neighbor nodes directly connected to Stanford node, but there are only 4 neighbor nodes directly connected to UCSC node. The decreased number of neighbor nodes is due to the edge trimming process. Edge-disjoint paths are the maximum number of paths that don’t have common edges. The results make sense, because from the zoom-in road mesh of source and the destination we can see that there are 6 neighbor nodes directly connected to Stanford node, but there are only 4 neighbor nodes directly connected to UCSC node. So the maximum number of paths that don’t have common edges should be a number that is <= 6.

|  |  |
| --- | --- |
| Stanford neighbourhood (-122.18, 37.43)  ID in Geo data = 2607 | UCSC neighbourhood (-122.06, 36.97)  ID in Geo data = 1968 |
|  |  |