EE 232E Project 5
Graph Algorithms

Hengjie Yang, Sheng Chang, Wandi Cui, and Tianyi Liu

June 15, 2018

1 Stock Market

1.1 Return Correlation

Question 1: Provide an upper and lower bound on ρ_{ij}. Also, provide a justification for using log-normalized return $r_i(t)$ instead of regular return $q_i(t)$.

According to the formula of ρ_{ij}, the upper bound should be 1 and the lower bound should be -1, where $r_i(t) = r_j(t)$ and $r_i(t) = -r_j(t)$ respectively.

Considering the reason why we use log-normalized return $r_i(t)$ instead of regular return $q_i(t)$, we have following analysis and explanations:

(a)log-normality: if we assume that prices are distributed log normally (which, in practice, depends on the given price series), then $r_i(t)$ is conveniently normally distributed, because:

$$r_i(t) = \log(1 + q_i(t)) = \log \frac{p_i(t)}{p_i(t-1)}$$

This is handy given much of classic statistics presumes normality.

(b)Approximate raw-log equality: when returns are very small (common for trades with short holding durations), the following approximation ensures they are close in value to raw returns:

$$r_i(t) = \log(1 + q_i(t)) \approx q_i(t), \text{where } q_i(t) \ll 1$$

(c)time-additivity: Consider a stock i with an ordered sequence of n trades. A statistic frequently calculated from this sequence is the compounding return, which is the running return of this sequence of trades over time:

$$q_i(1) \cdot q_i(2) \cdot q_i(3) \cdot ... \cdot q_i(n)$$

This formula seems really complicated since the product of normally-distributed variables is not normal. Instead, the sum of normally-distributed variables is normal. This can show another benefit of representing returns in log-normality:

$$r_i(1) + r_i(2) + r_i(3) + ... + r_i(n) = \log p_i(n) - \log p_i(0)$$

Thus, the compound return over n periods is merely the difference in *log* between initial and final periods. In terms of algorithmic complexity, this simplification reduces $O(n)$ multiplications to $O(1)$ additions.

(d)For Mathematical Ease: from calculus, we are reminded (ignoring the constant of integration):

$$e^x = \int e^x dx = \frac{d}{dx} e^x = e^x$$

This identity is tremendously useful, as much of financial mathematics is built on continuous time stochastic processes which rely heavily on integration and differentiation.

(e)Numerical stability: Sometimes the addition of small numbers is numerically safe, while multiplying small numbers is not as it is subject to arithmetic underflow.

1.2 Constructing Correlation Graphs

Question 2: Plot the degree distribution of the correlation graph and a histogram showing the un-normalized distribution of edge weights.

After constructing the correlation graph, the degree distribution we get is shown in Figure1 Not surprisingly, all vertices are connected with each other, so the degree of them are the same, equals to the number of vertices minus 1.

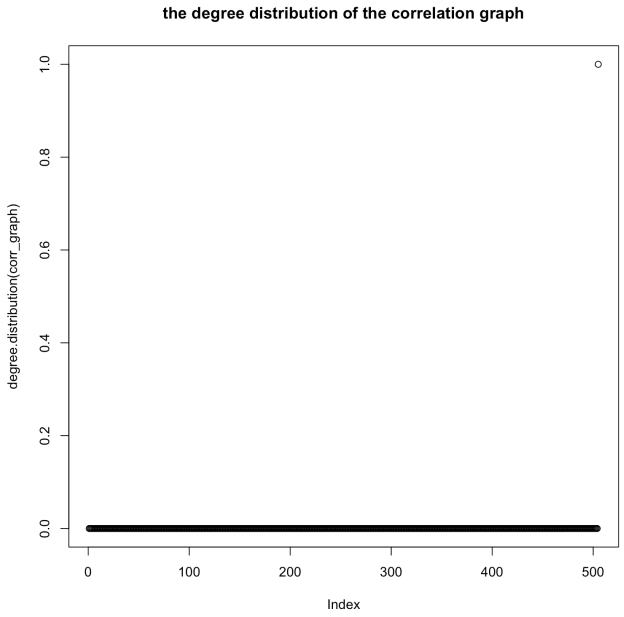

Figure 1: degree distribution of the correlation graph

The edge weights distribution is shown in Figure2

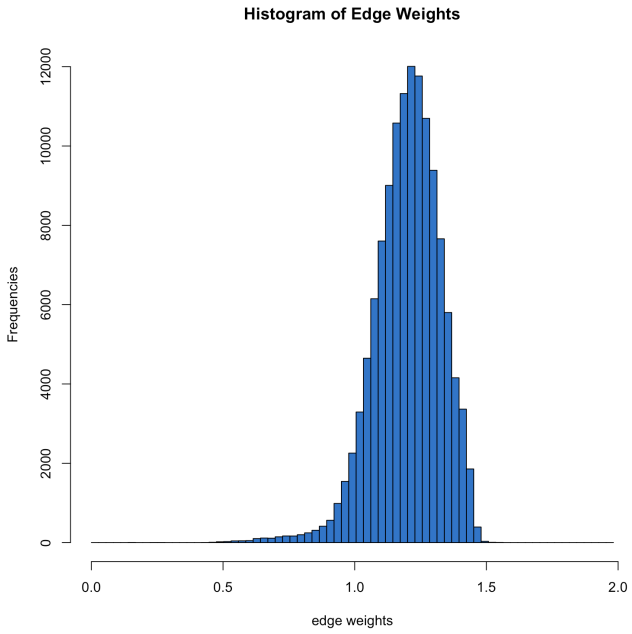

Figure 2: edge weights distribution of the correlation graph

1.3 Minimum Spanning Tree (MST)

Question 3: Extract the MST of the correlation graph. Each stock can be categorized into a sector, which can be found in Name_sector.csv file. Plot the MST and color-code the nodes based on sectors. Do you see any pattern in the MST? The structures that you find in MST are called Vine clusters. Provide a detailed explanation about the pattern you observe.

The MST plot is shown in Figure3, and differnt color represents different sectors of the stock.

We found that there is an interesting pattern in MST: stocks in same sector tend to be clustered together, and the whole MST looks like a grape-shaped graph. In fact, this is understandable, since the weight of the edge represents the correlation of the stock. It is not difficult to imagine that stocks are highly correlated so the weight between them is likely to be comparablly smaller than the weight between stocks in different sectors. Since MST find the path containing all the vertices in the graph with minimum total cost, it is reasonable to traverse nearly all the vertices in one sector and than transfer to another. That is why the pattern of MST looks like several "sector clusters" hanging in the tree, which is called "vine clusters" structure.

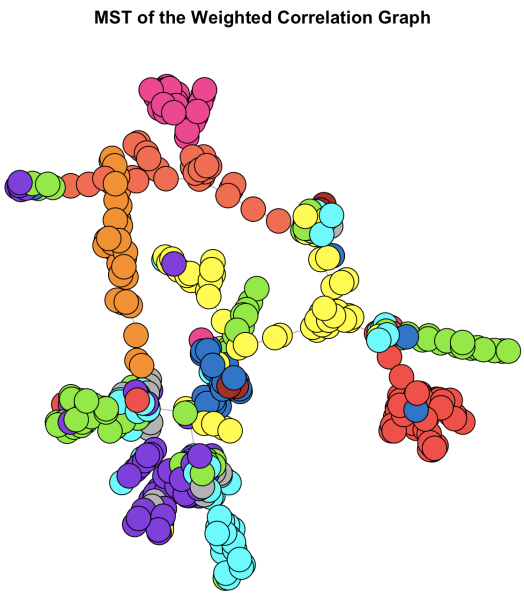

Figure 3: MST of the correlation graph

1.4 Sector Clustering in MST's

Question 4: Report the value of α for the above two cases and provide an interpretation for the difference.

Answer: case1: $\alpha = 0.82893$ case2: $\alpha = 0.11419$

To predict the market sector of an unknown stock, two methods for performing the task are utilized and compared. In case1, $P(v_i \in S_) = |Q_i|/|N_i|$, where Q_i is the set of neighbors of node i that belong to the same sector as node i and N_i is the set of neighbors of node i. For Case2, $P(v_i \in S_i) = |S_i|/|V|$.

The difference between these two results is that in case 1, information of neighbors of node i being predicted is used to infer its sector. The higher probability that its neighbors belong to certain sector, the higher chance it belongs to that sector. In case 2, however, the node i being predicted just is assigned with the label based on the percentage each sector takes up overall. The larger size the sector is, the higher probability it belongs to that sector, which is a naive random guess approach (sampling idea).

1.5 Correlation Graph for Weekly Data

Question 5: Extract the MST from the correlation graph based on weekly data. Compare the pattern of this MST with the pattern of the MST found in question 3.

The MST we got is shown in Figure 4.

Comparing this MST of weekly data and the MST of daily data, we can see that when we consider a larger time interval (weekly compared to daily), the vertices(stocks) in the weekly graph from same sector seems not as clustered together as daily graph. In other words, daily stock closing price is more likely to be influenced by the sector a stock belongs to; however, the relation demonstrated weaker when considering the weekly data. This is in fact easy to understand since when time interval becomes bigger, there may be several other factors could affect the stock price like changes in market.

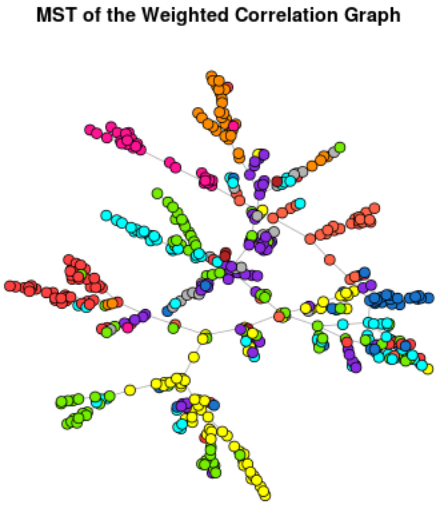

Figure 4: degree distribution of the correlation graph (weekly data)

2 Let's Help Santa!

2.1 Download the Data

2.2 Build Your Graph

Question 6: Report the number of nodes and edges in G.

After building the graph in which nodes represent the locations and undirected edges represent the mean traveling time,

(1) the number of nodes is 1880;

(2) the number of edges is 311802.

2.3 Traveling Salesman Problem

Question 7: Build a minimum spanning tree (MST) of graph G. Report the street addresses of the two endpoints of a few edges. Are the results intuitive?

Some examples of the street addresses are presented as in Table. 1. The results are very intuitive since the locations are very close to each other and the mean traveling time is very similar to the traveling time that the Google Maps provided.

Table 1: Examples of street addresses and mean traveling time

From	To	Mean traveling time (s)
3300 Brodie Drive, South San Jose, San Jose	4300 La Torre Avenue, South San Jose, San Jose	132.59
3300 Brodie Drive, South San Jose, San Jose	3700 McLaughlin Avenue, South San Jose, San Jose	126.24
1700 Coyote Point Drive, Shoreview, San Mateo	1800 Helene Court, East San Mateo, San Mateo	80.985
1700 Coyote Point Drive, Shoreview, San Mateo	600 Lexington Way, Oak Grove Manor, Burlingame	111.885
1400 Calle Alegre, South San Jose, San Jose	April Trail, Almaden, San Jose	166.725
1400 Calle Alegre, South San Jose, San Jose	5600 Park Crest Drive, South San Jose, San Jose	151.19

Question 8: Determine what percentage of triangles in the graph (sets of 3 points on the map) satisfy the triangle inequality. You do not need to inspect all triangles, you can just estimate by random sampling of 1000 triangles.

After the random samplings of 1000 triangles on the graph, the percentage of triangles in the graph is 93.9%.

Question 9: Find the empirical performance of the approximate algorithm:

$$\rho = \frac{\text{Approximate TSP Cost}}{\text{Optimal TSP Cost}} \tag{1}$$

According to the 2-approximate algorithm analysis, we know

$$\rho = \frac{\text{Approximate TSP Cost}}{\text{Optimal TSP Cost}} \leq 1.5698 \tag{2}$$

Question 10: Plot the trajectory that Santa has to travel!

After implementing the 2-approximate algorithm, the trajectory that Santa has to travel is shown in Fig. 5.

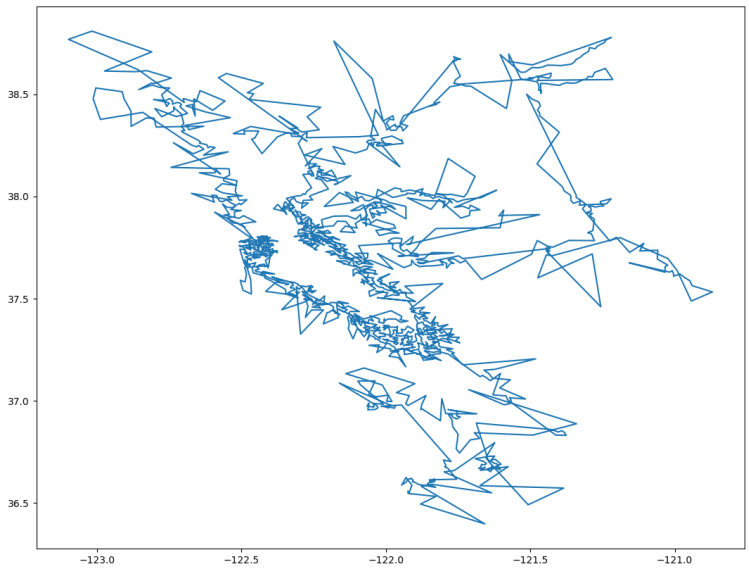

Figure 5: Travel trajectory that Santa has to travel

3 Analysing the Traffic Flow

3.1 Estimate the Roads

Question 11: Plot the road mesh that you obtain and explain the result. Create a subgraph G_Δ induced by the edges produced by triangulation.

After implementing the Delaunay triangulation, the road mesh we obtained is in Fig. 6. The result is quite intuitive as the density of the road mesh is proportional to the actual road density in that area. The road mesh is very similar to the actual map since more dots are needed to characterize the places with high transportation density. On the contrary, only a few number of dots are needed to characterize the wild areas.

3.2 Calculate Road Traffic Flows

Question 12: Using simple math, calculate the traffic flow for each road in terms of cars/hour.

We can calculate the traffic flow i.e. w cars/hour of each road by following equations.

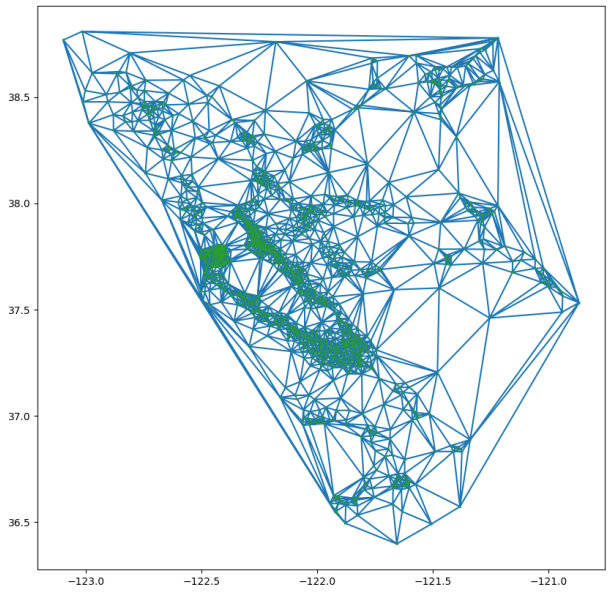

Figure 6: Road mesh after triangulation

$$w = 2 \times \frac{60 \times 60}{2 + \frac{0.003}{v}} \tag{3}$$

$$v = \frac{\text{euclidean distance} \times 69}{\text{mean time}} \tag{4}$$

Also, we notice that not all the road exist in the real word, which means we can not find the mean traveling times for those fake road. To solve this problem, we use shortest path calculation in case they are missing.

After going through all the node in the graph to calculate, the number of car of most roads is between 2000 and 3000 per hour.

3.3 Calculate the Max Flow

Question 13: Calculate the maximum number of cars that can commute per hour from Stanford to UCSC. Also calculate the number of edge-disjoint paths between the two spots. Does the number of edge-disjoint paths match what you see on your road map?

After calculation, the maximum number of cars that can commute per hour from Stanford to UCSC is 14892 and the number of edge-disjoint paths between them is 5, which seems like more than the real paths that I observe from the real Google map. As far as I am concerned, this phenomena can be explained by those fake roads we generate from the Delaunay triangulation.

3.4 Defoliate Your Graph

Question 14: Plot G_Δ on real map coordinates. Are real bridges pre- served?

In this part, we aim to delete these fake brights. There're no doubt that we can assume that these fake bridges are taking a really convoluted route to achieve the distance between two vertices, and hence their mean traveling times are going to be very higher. Therefore, we apply a threshold on the travel time of the roads.

The average time for a road is 362. However there're a little road' time even greater than 1000. Those ridiculous time means that those roads don't exist in the real world at all. So we can set 1000 as our threshold to prune those fake edges.

Question 15: Now, repeat question 13 for G_Δ and report the results. Do you see any significant changes?

After defoliating the road map, we find that there're almost no significant change in the result of max flow even after removal of the fake edges. As far as I am concerned, even though when those fake roads exist, those roads especially fake brights can be seen as the outlier edges for the path between Stanford and UCSC so that they almost do not contribute to the any flow at all.