

## Homework 4

Aozhu Chen 004773895  
Ruoxi Zhang 404753334  
Ning Xin 704775693

In this assignment, we will study data from the stock market. We study correlation structures among fluctuation patterns of stock prices. We construct different graphs based on similar among the time series of returns on different stocks at different time scales (day vs week).

### Question 1:

In this problem, we calculate the correlation among time series data. We use the equation below to calculate the cross-correlation coefficient of two different stock-return time series:

$$\rho_{ij} = \frac{\langle r_i(t)r_j(t) \rangle - \langle r_i(t) \rangle \langle r_j(t) \rangle}{\sqrt{(\langle r_i(t)^2 \rangle - \langle r_i(t) \rangle^2)(\langle r_j(t)^2 \rangle - \langle r_j(t) \rangle^2)}}$$

where  $i$  and  $j$  means two different stock and the log return of the closing price  $r(t)$  is calculated by the closing price of each stock each day. The equation is calculated by following equation:

$$r_i(t) = \log p_i(t) - \log p_i(t - \tau)$$

**we choose to use the log return of stock prices for following reasons:**

- it is time additive and time consistent
- Since The stock return are generally close to normal distribution, by using log prices we can convert an exponential problem to a linear problem.

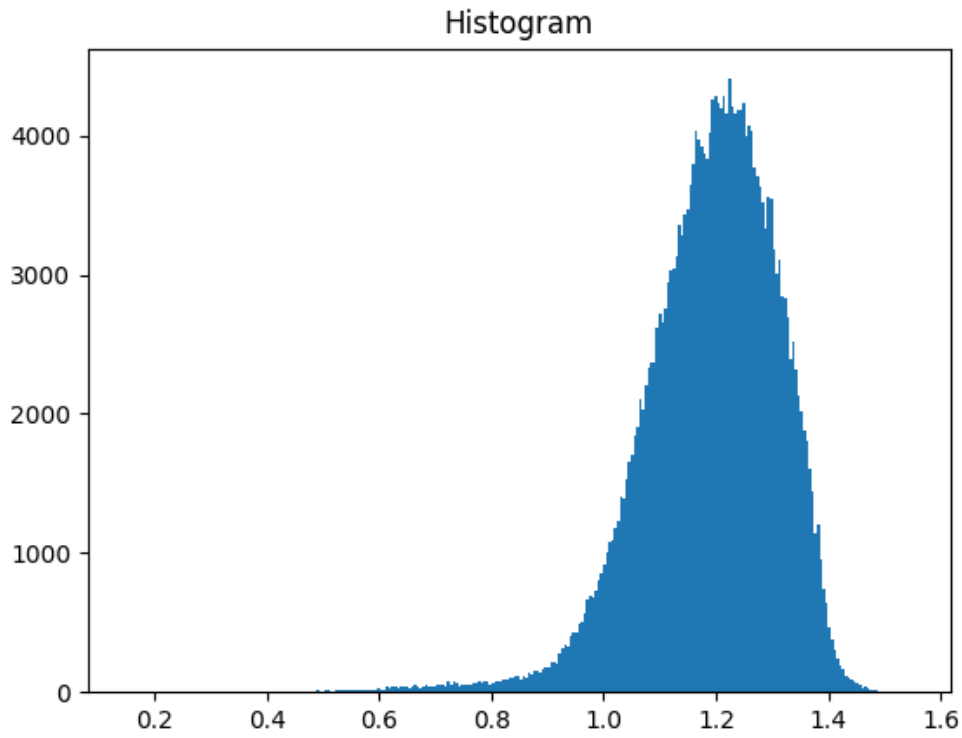
Also in this part, we exclude the stocks that have less than 765 rows. In that case **the total number of stocks we chose to use is 494. The value of  $\rho_{ij}$  ranges from -1 to 1.**

### Question 2:

In this problem, we construct correlation graph. The graph constructed is an undirected graph, the weight of edges is calculated by following equation for different stocks:

$$d_{ij} = \sqrt{2(1 - \rho_{ij})}$$

where  $i, j$  represent different stock. Each stock is represented by each vertex in the graph. The histogram of  $d_{ij}$ 's is a s following:



The weighted graph  $G$  is constructed by the adjacency matrix  $D=[d_{ij}]$ , The diagonal element of adjacency matrix is ignored to avoid the loop vertex.

The number of vertices = 494

The number of edges = 121771

**Question 3:**

In this problem, we find the minimum spanning trees for the correlation graph.

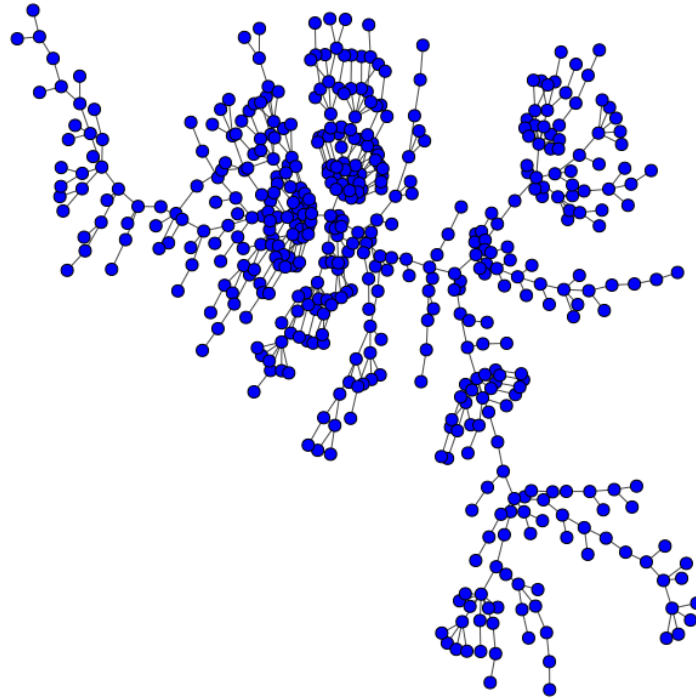


figure 3.1 the minimum spanning tree without color

From the uncolored minimum spanning tree for the correlation graph. We can see that some of the stocks tend to group together but we cannot find any more valuable information. The following graph is the MST with color. The color represents different sectors of the stocks in name\_sector.csv.

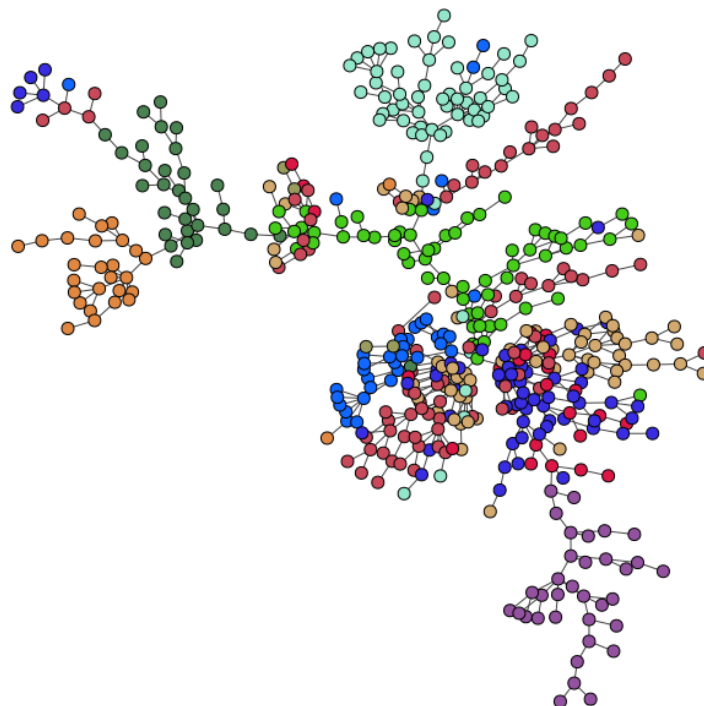


figure 3.2 the minimum spanning tree with color

From the graph, above, we can see that the most vertices with same color are closely connected to each other. This observation means that the stocks in same sector form the cluster in the MST.

#### Question 4 :

In this problem, we are trying to use the sector clustering features and the minimum spanning tree to predict the market sector of unknown stocks.

If we only take into account the immediate neighbors of a stock in the MST, we can evaluate the performance using below equation:

$$\alpha = \frac{1}{|V|} \sum_{v_i \in V} P(v_i \in S_i)$$

Here alpha stands for the performance coefficient,  $|V|$  is the total number of vertices,  $S_i$  is the sector of the node  $i$ , the neighbors of vertex  $i$  is  $N_i$ , the probability that  $v_i$  belongs to sector  $S_i$  is:

$$P(v_i \in S_i) = \frac{|\{j | v_j \in N_i, S_j = S_i\}|}{|\{j | v_j \in N_i\}|}$$

For the purpose of comparison, we also examine the case where each MST node is assigned a random sector. Which means the probability of a node being assigned the sector  $S_i$  is proportion to the percentage of this sector in the whole stock groups.

Then we implemented these 2 methods in the minimum spanning tree we generated in previous questions. The results are listed below:

```
performance: 0.828930077531
random performance: 0.112391291851
```

Figure 4-1 Performance Coefficient in 2 calculation methods

We can see that for the ordinary calculation method, the performance coefficient is about 0.83 while in random way this value is 0.11, which means that the ordinary method has a better accuracy than random one. So we can say that this is an effective approach. For further implementation details, please refer to the source code.

#### Question 5 :

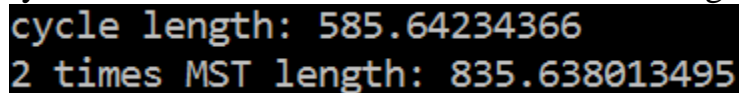
In this question, we are trying to clarify and solve the problem of Delta-Traveling Salesman Problem, in which we want to find a route to traverse every vertex without visiting any vertex more than once. The way we approach this problem is to use minimum spanning tree to approximate the length of salesman's route. This approach has a basic assumption, which is that the graph should obey triangle inequality. However, in our task, the graph we use is fully connected but not necessarily obeys triangle inequality. So first we need to prove that this inequality holds for fully connected graph.

We prove it by contradiction. Now that we know the graph is fully connected, which means there must exist a shortest path between each pair of vertices. So for an arbitrary set of 3 nodes  $x, y, z$ ,  $d(x, y)$  stands for the shortest path between vertex  $x$  and  $y$ . Suppose  $d(x, y) + d(y, z) < d(x, z)$ . By definition,  $d(x, y)$  is the length of the shortest path from  $x$  to  $y$ , and  $d(y, z)$  is the length of the shortest path from  $y$  to  $z$ . However,  $d(x, z)$  is the length of the shortest path from  $x$  to  $z$ , so  $d(x, y) + d(y, z)$  can't be less than  $d(x, z)$ . So we have a contradiction.

Now we proved this assumption, we can then use minimum spanning tree (MST) to approximate TSP. First, we start from a randomly picked node in MST, then we double the edges between every pair of nodes, so that now we can think that between each pair of connected nodes in MST, there are 2 identical edges between. Then we will start traverse the node following the spanning tree edges, at the meantime we mark every node as visited. Then when we in the situation where the node we are going to visit has already been visited, then we will directly jump to next unvisited node following the original edge in graph. And we keep doing this operation until we traverse all the nodes in MST.

Then according the triangle inequality of graph, we can find that the length we calculated by the above algorithm is guarantee to be in the range of 1 MST and 2 MST. What needs mentioning is that the visiting order matters, but even though the total length maybe different but the result is strictly bounded by 1MST and 2MST.

Below figure is the execution result of stock relation graph. The cycle length we calculated is 585.64, while 2 times of MST length is 835.64, which match our algorithm properly. And the error rate is about 40% of MST length.

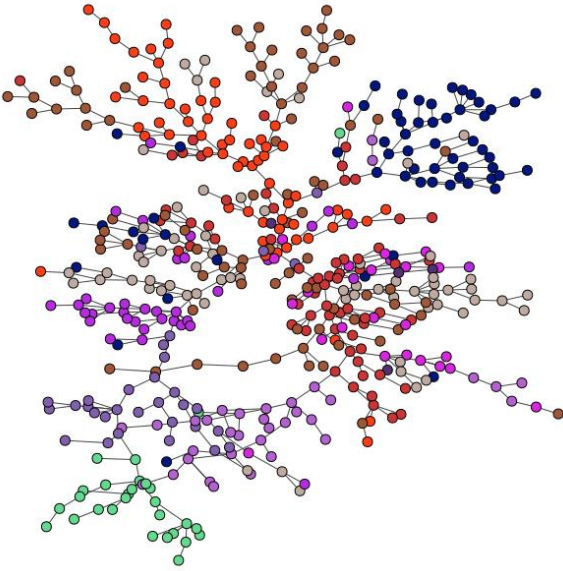


```
cycle length: 585.64234366
2 times MST length: 835.638013495
```

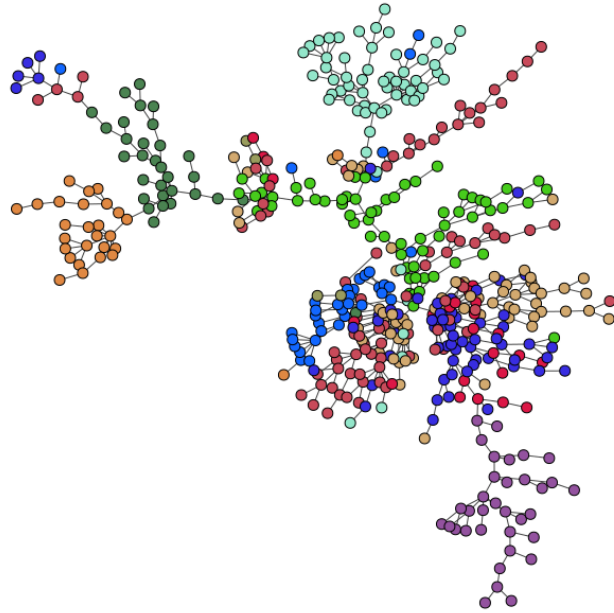
Figure 5-1 Execution Result of MST Approximation Algorithm

### Question 6:

In this problem, we construct correlation graph using weekly data on Mondays. The graph constructed is an undirected graph, the weight of edges is calculated by the equation in the question1. The MST and dij is construct as before. The result is as following:



**Figure 6.1 the MST using weekly data**



**Figure 6.2 the MST using daily data**

As we can see from the structure of the MST is different. However, the pattern of the MST remains the same. The stocks in same sector are closely connected to each other. However, it is obvious that the MST in previous sample data has much closer cluster.

### Question 7:

In this problem, we construct correlation graph using weekly data on Mondays. First we plot the histogram of  $\rho_{ij}$  from daily data. The result is as following:

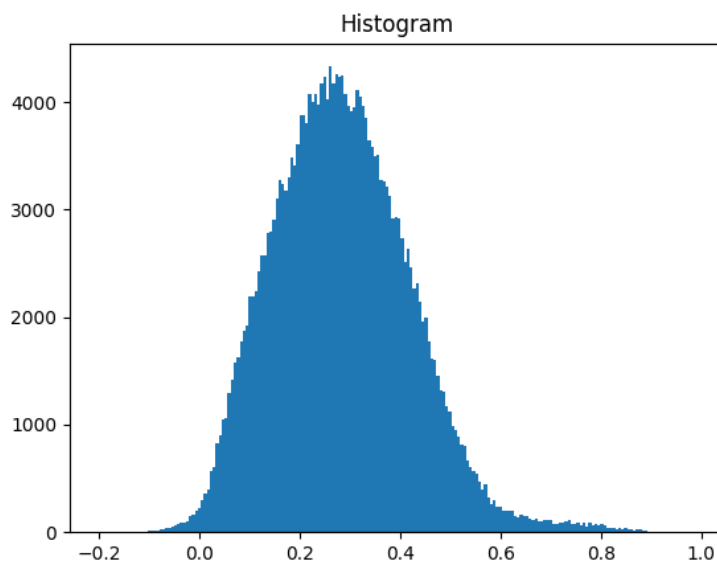


Figure 7.1 the histogram of  $\rho_{ij}$  from daily data

Then, we modify the correlations matrix. We use the new correlations matrix to construct the graph and run MST. The rules we applied here is as following:

- $\rho_{ij}$ s larger than 0.3 are set to -1
- $\rho_{ij}$ s less than and equal to 0.3 remain same

The result is as following:

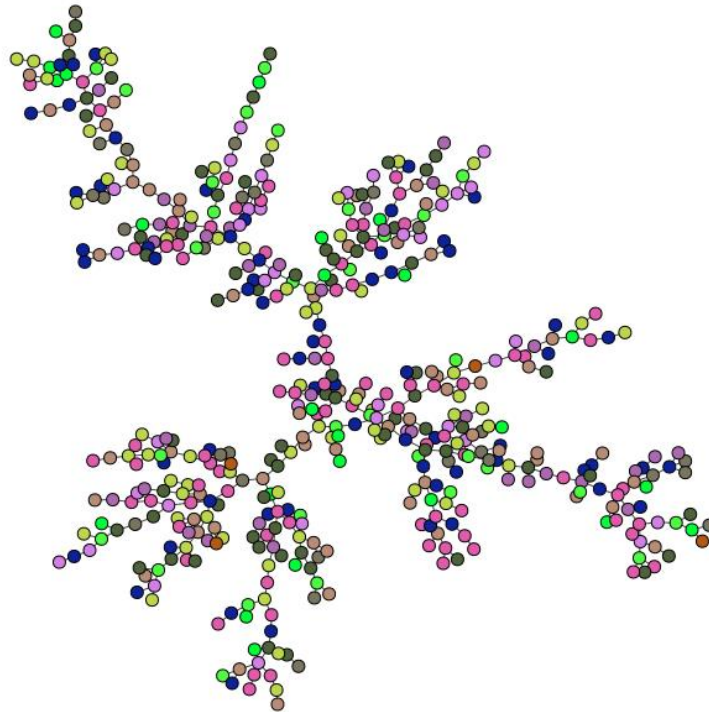


Figure 7.2 the MST using weekly data and new Correlations

As we can see from the figure above, the stocks in the same sector no longer group together. The vine cluster no longer exists. This makes sense because the vertices with strong correlations (larger than 0.3) are set to -1 therefore the edge weight between these two vertices becomes very large. Therefore, the edge between the stocks in the same sector no longer preferred when we construct the MST.

### Question 8:

The generative model we come up is as following:

1. Assign tag "1" to two nodes
2. Generate a new node, attach the node to one of the nodes exists with some probability. The probability is proportional to the node value
3. Update the tag value. The new node becomes one. Change the value of the node been attached to  $1/\text{deg}$  of the node
4. Repeat 1 – 3 until the number of node is large enough.