CPTS 515 Advanced Algorithms Final

Ivani Patel
Washington State University

Abstract— **Predicting the price correlation of two assets for future time frames is significant in portfolio advancement. This research paper aims at predicting stock market based on previous days' data. Stock prediction aims to predict what the future trends of a stock holds in order to help investors to make good wise endeavor decisions. This paper also aims to measure the similarities between two stock charts. This two question will be answered using the knowledge of hashing, Markov chain, de Bruijn graph and matrices.**

Index Terms— **Stock market prediction, Markov chain, de Bruijn graph, hashing, matrix**

I. INTRODUCTION

Stock market is one of the significant fields that investors are committed to, subsequently stock market price trend prediction is always a hot topic for researchers from both financial and technical domains. People tends to believe that stock market is unpredictable. In that case, if we closely look at a stock chart, we will find the data on the chart should be predictable since they are not randomly generated at all. However long we have an adequate number of specific information of one stock, we could figure out a way to predict the pattern of that stock. Which will help both traders and investors to make decisions to decisions to either buy or sell an underlying asset which could yield significant profit. This paper aims to answer two questions:

(A) How to predict tomorrows' data for a stock based on previous chart history.

(B) Given two stack charts, how to determine the similarity between those two.

This report documents the hashing, de Bruijn graph, Markov chain and comparison of matrices to predict the pattern of stock and to determine the similarity between two stock charts. This paper is structured as follows: Section II provides detailed approach to predict one stock chart. In Section III the details about tracking similarities between two charts has been provided; followed by conclusions and references.

II. APPROACH TO PREDICT ONE STOCK CHART

Step 1: Problem set up and analysis

Using A(α) to denote a stock predicting algorithm A which takes only one input α where α is a stock chart. A stock chart α has a very simple data structure: an array $\alpha[1...k]$ for some large k. Each element $\alpha[i]$ is a market data for the stock of day i, which is represented by a tuple of five numbers:

- Open price of the day
- Close price of the day
- Highest price of the day
- Lowest price of the day
- Number of shares traded in the day(volume)

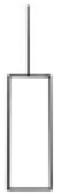
Step 2: Hash this array

We can not track down a clear connection from graph to decide whether we'll get profit or we'll lose money and the value of volume. In the occasion volume really impact the price of the day, we generally divide volume in 9 categories which are in million to simplify review.

[0, 50], (50, 100], (100, 150], (150, 200], (200, 250], (250, 300], (300, 350], (350, 400], (400, +∞]

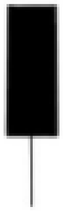
Our goal is to gain profit, to make money, and that implies we will purchase this stock when we anticipate the "cost" for the following day is higher than today. Considering every single days' price trend, we divide them into the following 5 categories:

1. open price = the lowest price

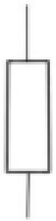


In this case, "price" of following day will be completely higher than the present. One will thoroughly acquire money if they hold this stock today, say 100% win.

2. open price = the highest price



In this case, the "price" of following day will be absolutely lower than the present. One will absolutely lose cash if they hold this stock today, say 0% win.



3. close price > open price but none of them is either highest or lowest

In this case, the price of following day will be in some cases lower than today, yet taking everything into account, the price of following day will be higher than today. The probability we will earn money in this day can be computed as:

$$Prob = \frac{highestprice - openprice}{highestprice - lowestprice}$$

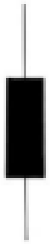
Then we set 5 probability domains to classify every probabilities under this case:

$$(0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1)$$

For each range, we use the middle number(average number) to denote the win rate which can be represented by:

$$0.1, 0.3, 0.5, 0.7, 0.9$$

4. open price > close price but none of them is either highest or lowest



Taking everything into account, will lose money at this day. The win rate can be figured by:

$$Prob = \frac{highestprice - openprice}{highestprice - lowestprice}$$

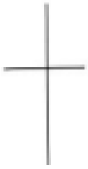
Similarly, Then we set 5 ranges to classify every probabilities under this case:

$$(0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1)$$

For each range, we use the middle number(average number) to denote the win rate which can be represented by:

$$0.1, 0.3, 0.5, 0.7, 0.9$$

5. open price = close price



The win rate can be computed by:

$$Prob = \frac{highestprice - openprice}{highestprice - lowestprice}$$

Similarly, Then we set 5 ranges to classify every probabilities under this case:

$$(0, 0.2], (0.2, 0.4], (0.4, 0.6], (0.6, 0.8], (0.8, 1)$$

For each range, we use the middle number(average number) to denote the win rate which can be represented by:

$$0.1, 0.3, 0.5, 0.7, 0.9$$

As per the above classifications, we can observe that success rate of all circumstances can be determined by:

$$Prob = \frac{highestprice - openprice}{highestprice - lowestprice}$$

Now resetting classes into 7, considering win probability:

- 0% when Prob = 0
- 10% when Prob ∈ (0, 0.2]
- 30% when Prob ∈ (0.2, 0.4]
- 50% when Prob ∈ (0.4, 0.6]
- 70% when Prob ∈ (0.6, 0.8]
- 90% when Prob ∈ (0.8, 1)
- 100% when Prob = 1

Up to this point, we have absolutely 9 * 7 = 63 classifications, and we hash the chart α[1 ...k] into those 63 slots. The quantity of classifications can be adjusted by changing the range of probability domains.

Step 3: Construct the de Bruijn Graph

The hash table is restricted, so we expect to such an extent that hash table is a limited symbol set. Resulting to hashing α[1 ...k], each part $\alpha[i]$ can be represented by a symbol. For each symbol, there is a probability can be figured which is the rate we can acquire money. Then, at that point, we can get a sequence of symbols which can address the graph.

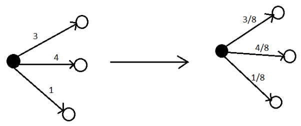
In view of this succession, we build the de Bruijn graph. First we want to pick a positive boundary, say n = 4.

Pick the initial 4 symbols in the arrangement as one node in de Bruijn graph, recursively, move backward one step at a time, each time consolidate 4 symbols to 1 node in the graph. At the same time, add directed edges between nodes considering the order of sequence.

To switch it over completely to Markov chain, we put a count m to each edges, where m is the quantity of appearance of substring in the original sequence. For this situation, the length of substring is 5.

Step 4: Normalization

Normalize the de Bruijn graph into a Markov chain. We really want to cause the amount of probabilities on every node to be 1. For instance, one node in de Bruijn diagram can be standardized as follow:



Now we develop the transition probability matrix T.

$$T_\alpha = [P_{ij}] = \begin{bmatrix} P_{1,1}...P_{1,n} \\ . \\ . \\ . \\ P_{1,1}...P_{1,n} \end{bmatrix}$$

where P_{ij} implies the probability from station P_i to P_j, also, the summation of each and every line is 1.

Up to this point, as long as we have the chart of 2000 previous days' information and the present station (symbol), we can know the probabilities of going different next stations with the success rates. Then, we can for the most part anticipate the cost for the following day.

III. Approach to Determine Similarity Between Two Charts

In the previous section, we realize that we can hash one stock chart, and subsequently use Markov chain to assemble the transition probability matrix T. For every stock chart, we will get a unique transition matrix. In this manner, the question of determining the similarity between two charts can be converted into determining the similarity of two transition probability matrices of charts.

The likeness between two matrices can moreover be considered as the distance between two matrices.

Expect that we have matrix A which we get from chart α and we have matrix B which get from chart β. Now, we really want to determine the distance between A and B.

Transition matrix should be square, so one method for computing the distance between (A,B) is utilizing Frobenius distance F [1]:

$$F_{A,B} = \sqrt{trace((A-B)(A-B)')}$$

where B' addresses the conjugate transpose of B, and the trace of matrix is the sum of the diagonal elements. The smaller $F_{A,B}$, the more equivalent the two matrices.

There are additionally another ways of deciding the distance between two matrices:

$$d_1(A,B) = \sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij} - b_{ij}|$$

$$d_2(A,B) = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} (a_{ij} - b_{ij})^2}$$

$$d_\infty(A,B) = \max_{1\le i\le n} \max_{1\le j\le n} |a_{ij} - b_{ij}|$$

For all of above techniques, the smaller d(A,B), the smaller distance between two matrices, the two matrices are more similar, the comparability between two charts are higher.

IV. Conclusions

In this paper, we analyzed one strategy to predict the price of the following day with the input is a single chart, and a couple of strategies to decide the similarity between two stock charts. For the first question, accepting we have the stock blueprint for each day and the expense for as a matter of course, we can use integral to calculate the extent of area, the probabilities we get will be more accurate. For instance, one day's close price is a lot of lower than open price, we really could acquire money at that day because at some point in time, the price can be extremely high. Subsequent to building a suitable hash function and hashing stock chart, the stock chart can be addressed by a finite symbol sequences, create de Bruijn graph out of this sequence, normalize it into Markov chain, and construct transition probability matrix. Up until this point, we can predict the price for the following day in view of the transition matrix. In the subsequent inquiry, translating original into determining the similarity between two matrices then, at that point, we related "distance" with "closeness". There are numerous ways of managing measure the distance between two matrices, the inquiry is whether we can consider "distance" and similitude as the equivalent.

References

[1] Distance/Similarity between two matrices [https://math.stackexchange.com/questions/507742/distance-similarity-between-two-matrices.]

[2] Markov Chain [https://en.wikipedia.org/wiki/Markov_chain]

[3] Markov Chain Applied to Returns on Stock Prices https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3645482

[4] Compacting de Bruijn graphs from sequencing data quickly https://academic.oup.com/bioinformatics/article/32/12/i201/2289008