Stock Data Prediction with Vector Grouping and Markov Modeling

Thitaree Tanprasert
Harvey Mudd College
Claremont, CA 91711
ttanprasert@hmc.edu

Julianne Lin
Harvey Mudd College
Claremont, CA 91711
jullin@hmc.edu

April 2, 2019

Abstract

In this project, we perform an analysis on time-series stock data to extract patterns and find correlations between the stock data and other economic, social, and political phenomena. We propose an algorithm that differs from any previous research on this topic. Instead of looking at the raw data as is, we develop an algorithm to describe each dip and peak in the data in terms of rotation angle, scaling factors, and shearing factors with respect to a normalized vector. Then, we categorize similar vectors together. Then, we use Markov model to predict vector patterns based on this categorization. The system prediction accuracy is at 18%. However, there is a potential for improving the performance because this feature extraction scheme can reduce dimensions of the input stock data and extract meaningful characteristic of the data. We plan to optimize the algorithm and the Markov model in the future to determine the effectiveness of vector grouping in time-series data prediction in general.

1 Introduction

Predicting stock trends is a widely explored problem in the financial sector. There have been numerous attempts at learning the pattern solely from the numerical time-series data. But the change in stock data also relies on non-qualitative phenomena, and the knowledge of the relationships between stock data and relevant ongoing issues, including other social and political changes, maybe crucial in the accuracy of prediction.

In this project, we perform an analysis on time-series stock data to extract patterns and find correlations between the stock data and other economic, social, and political phenomena. In order to achieve this, we develop a scheme to model data patterns using vector transformation. More specifically, we develop an algorithm to describe each dip and peak in the data in terms of rotation, scaling, and shearing of a normalized V-shaped vector - a shape formed by alternation of local maxima and minima that occurs repeatedly in any time-series data. By extracting and analyzing these V-shaped structures in the data, we hope to obtain an insight at a higher level than merely numerical patterns. Then, we perform prediction based on these extracted features.

This paper consists of 7 sections, including this introduction. In the second section, we summarize previous research that has explored the stock data prediction problem. In the third section, we explain the proposed approach for vector grouping. In the fourth section, we present the predictive model we use to test the effectiveness of our vector grouping algorithm for stock data prediction. In the fifth section, we describe the details of our experiments, including the properties of the dataset we use and the system implementation. In the sixth section, we present and interpret our experiment results. Finally, we discuss the current state of our work as well as the plan for future work in the seventh section.

2 Literature Review

Similar work done previously to predict stock trends focus on other methods of time-series analysis and stock-predictions. To extract features for a predictive model, current literature has used supervised learning methods like kernel ridge regression [3] as well as unsupervised feature extraction algorithms like orthogonal wavelet transforms [4]. Other approaches include varying the feature sets of the data, like using a combination of S&P500 data and the data of the highest-priced target company covering the past five days [5]. In implementing the predictive model, previous research groups have used unsupervised learning models like K-nearest neighbors, Naive Bayes classification, tree-based classification, and SVM (support vector machines) [1]. Other techniques include supervised learning and hybrids approaches, like recurrent neural networks and a combination of ridge regression and genetic algorithms, respectively [2,3].

Many of these past works rely on understanding characteristics of the stock market and other possible influencing economic factors to model their data. These methodologies are often fine-tuned to apply specifically to stock data predictions and analyses. Our approach is generalized to any time-series data and independent of the characteristics specific to the stock market. We hope to verify the validity of our method using stock data with the potential to extend it to apply to any time-series data.

3 Vector Grouping Methodology

The vector grouping methodology section will consist of three main subsections: vector extraction, transformation parameters computation, and parameters grouping.

3.1 Vector extraction

The input to the system is discrete time-series data, where each timestamp corresponds to a value and the timestamps are equally spaced. In case of stock data, the timestamps are the dates in which the stock price is recorded. We would like to transform this input data into a sequence of V-shape vectors, starting from the first local maximum in the data.

In order to do this, we retrieve the x, y-coordinates of all local minima and maxima. The data between two consecutive local maxima constitute a single V vector. Each V vector will be represented by the two local maxima at the end and one local minimum that is the dip of the V-shaped vector.

$$V = \begin{bmatrix} x_m - x_l & x_r - x_m \\ y_l - y_m & y_r - y_m \end{bmatrix},$$

where (x_l, y_l) = timestamp and corresponding value of the left end of the vector

(x_m, y_m) = timestamp and corresponding of the local minimum

(x_r, y_r) = timestamp and corresponding of the right end of the vector

3.1.1 Transformation parameters computation

Transformation function computation consists of two steps. The first one is computing transformation matrix. That is, for a given pair of vectors V extracted from the data, we want to find a 2x2-matrix A such that $V = AH$. We can easily see that

$$A = VH^{-1} = \frac{1}{2}\begin{bmatrix} -V_{11} + V_{12} & V_{11} + V_{12} \\ -V_{21} + V_{22} & V_{21} + V_{22} \end{bmatrix}$$

The next step is transformation matrix decomposition, where we will decompose A into a product of three 2x2 matrices called R, S and SH:

- R corresponds to rotation, defined by an angle . To rotate by angle θ, counterclockwise, we multiply the vector by the matrix

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- S corresponds to scaling, defined by two scalars s_l and s_r for scaling the magnitude of the left and right vectors, separately. That is we would like to find S such that

$$S * H = \begin{bmatrix} -s_l & s_r \\ s_l & s_r \end{bmatrix}$$

Therefore, we get

$$S = \frac{1}{2}\begin{bmatrix} -s_l + s_r & s_l + s_r \\ -s_l + s_r & s_l + s_r \end{bmatrix}$$

- SH corresponds to shearing parallel to the x-axis, defined by a scalar m. To shear a plane, we multiply the vector by the matrix

$$SH = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix}$$

We can decompose A by calculating the product of the matrices listed above

$$\begin{aligned} A &= R * S * SH \\ &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} -s_l + s_r & s_l + s_r \\ -s_l + s_r & s_l + s_r \end{bmatrix} * \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} s_l \cos\theta & m s_l \cos\theta - s_r \sin\theta \\ s_l \sin\theta & m s_l \sin\theta + s_r \cos\theta \end{bmatrix} \end{aligned}$$

From this, we can calculate the parameters θ, s_l, s_r and m as follow:

$$\begin{aligned} \theta &= \arctan(A_{21}/A_{11}) \\ s_l &= A_{11}/\cos\theta \\ s_r &= A_{22} - A_{12}\sin\theta \\ m &= (A_{12} + s_r \sin\theta)/s_l \cos\theta \end{aligned}$$

Each V-shaped vectors could therefore be represented by a tuple (θ, s_l, s_r, m). This makes it easier for us to categorize changes in the stock data, since we can do it with respect with each of the three parameters, independently.

3.2 Parameters grouping

We would like to group the V-shaped vectors together based on their transformation with respect to the normalized vector. For simplicity in this midterm project, we will only use the parameters s_l and s_r to group vectors together. To do this, we calculate the mean value of both parameters, $\bar{s_l}$ and $\bar{s_r}$. Then, we categorize the vectors by quotient of the vector's scaling parameters and the mean of that parameters. In other words, $V = (\theta, s_l, s_r, m) \in G_{l,r}$ where l $\frac{s_l}{\bar{s_l}}$ and r $\frac{s_r}{\bar{s_r}}$. $G_{l,r}$ is a single integer label for vector V. We can now represent our original stock data as a sequence of these labels. This process to extract meaningful features and significantly reduce the complexity of the data.

4 Predictive Model

In this project, we use k-order Markov model to learn and predict the extracted features of stock data. Considering the group labels of the vectors as possible

states, we can predict the next state (state $n+1$) by looking at k preceding states according to the formula

$$P(x_n|x_{n-1}, x_{n-2}, ..., x_1) = P(x_n|x_{n-1}, ..., x_{n-k})$$

The prediction output is chosen "randomly" according to the probability of possible states for the given input. It should be noted that the prediction for the same input will not be the same every time.

We can use the prediction of state $n+1$ to predict state $n+2$ (based on real data x_{n-k+1} to x_n and predicted data x_{n+1}) and iterate this to produce a sequence of prediction that is a Markov chain.

Theoretically, the higher the order of the model, the more context we have for our prediction and the more accurate our prediction should be. However, increasing the order of the model also increase the training time significantly. Let the number of possible states be N, the size of probably table for k-order Markov model is $O(N^{k+1})$. Therefore, in our experiment, we only use model of order 3. It should be noted that, based on the preliminary result from the subset of our data, we find that increasing the order to 5 and 7 does not seem to increase the accuracy of the prediction. We conjecture that this could be because, as the complexity of the model increases, the number of data points stays the same. In other words, we do not have enough data points to train a more complicated model. This will be discussed further in the experiment and result sections.

5 Experiment

The experiment section consists of two subsections: dataset and implementation. Specifically, we will describe the datasets we use or plan to use in this project and the details of our implementation of the algorithms described in sections 3 and 4.

5.1 Datasets

The dataset we use to experiment our method is the S & P500 stock dataset [1] which includes 5 years of historical stock data (February 8, 2013 - February 7, 2018) from 500 S& P companies. The dataset contains 619,040 data points (one for each day.) In each data point, we get the following attributes of the stock data:

- Open - Price of the stock at market open (this is NYSE data so all in USD)
- High - Highest price reached in the day
- Low - Lowest price reached in the day
- Close - Price of the stock at market close

[1] https://www.kaggle.com/camnugent/sandp500

- Volume - Number of shares traded
- Name - the stock's ticker name

Another dataset that we will potentially experiment on later is the Huge Stock Market Dataset [2] which is much larger than the previous dataset. The data provided in this dataset contains the same attributes as in the S & P500 dataset.

5.2 Implementation

We implement the system in Python 3, using numpy and scikit-learn packages to implement the Markov model and perform the K-fold validation, respectively. The Python jupyter notebook for running the experiment can be found online. [3]

6 Result

6.1 Data analysis and visualization

After we categorize each vector, we plot the data again and color code the data points to gain more insights into the distinctive quality between groups of vectors. Using the S & P500 data, we classify data into 15 groups. However, 6 groups occur at much higher frequencies in comparison to the other 9 groups as shown in Fig 1. Specifically, group 3, 4, 6, 7, 9, and 10 are much more abundant than the rest.

We assign a unique color to each of the group and plot random sections of our data for analysis and validation purpose. Fig 2 show two examples of visualization. Each of them represent data of length 80-90 days. We can see that the red vectors have similar characteristics with short left vectors and large right vectors. The blue vectors have similar characteristics but both left and right vectors are at larger scales. On the other hand, the green vectors have larger left vectors compared to the right. The magenta vectors are larger than the rest in general, and the relative length of left and right vectors are also similar to each other.

6.2 Prediction accuracy

We separate the data into training and validation datasets with ratio of data points 80:20. The training dataset contains 119,030 data points and the validation dataset contains 31,444. We use a third-order Markov model, so each data point is a 3-dimension list. The data point is a sequence of 3 consecutive V-shaped vectors $[V_{n-2}, V_{n-1}, V_n]$ where V_n represents the integer label of the n-th V-shaped vector in the data. It should be noted that one V-shaped vector could span multiple days of stock data, hence we have significantly less number of data

[2] https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs
[3] https://github.com/ttanprasert/MATH189SP19/blob/master/Midterm/Midterm%20Experiment%20(Final).ipynb

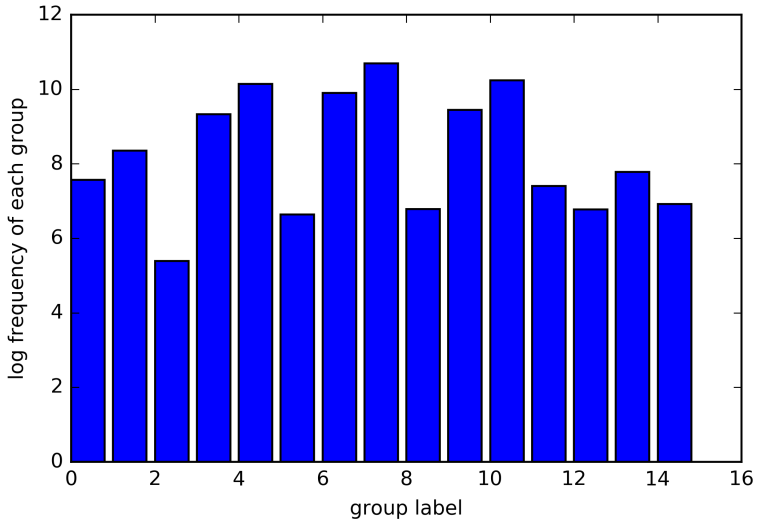

Figure 1: Log of frequencies of data points from S & P 500 dataset in each group

points here compared to the number of data points in S & P 500. For each data point, we predict 3 steps into the future. That is, we obtain the prediction in the form $[V_{n+1}, V_{n+2}, V_{n+3}]$.

Because the prediction process of a Markov model is not deterministic, we re-run the experiment with 5-fold cross validation. For each fold, we run 3 experiments. We find that the average accuracy of prediction is 17.94% for all 3 steps with standard deviation of 0.32%.

7 Conclusion and Future Work

In this project, we find that vector grouping is potentially an effective way of categorizing time-series data, as we can see from section 6.1. However, the experiment with Markov model, although gives a prediction of higher accuracy than random answers, still performs much more poorly than we expected. Moving forward, we aim to improve the prediction accuracy as follow:

- We would like to improve the grouping algorithm for based solely on the two scaling factors, such that the algorithm maintains the low number of groups but distributes the number of vectors in each group more evenly.
- We will expand upon the result we get for the scaling factors and develop a grouping algorithm that takes rotation angle and shearing factor into account as well. We hypothesize that this additional information will make our prediction model more accurate.

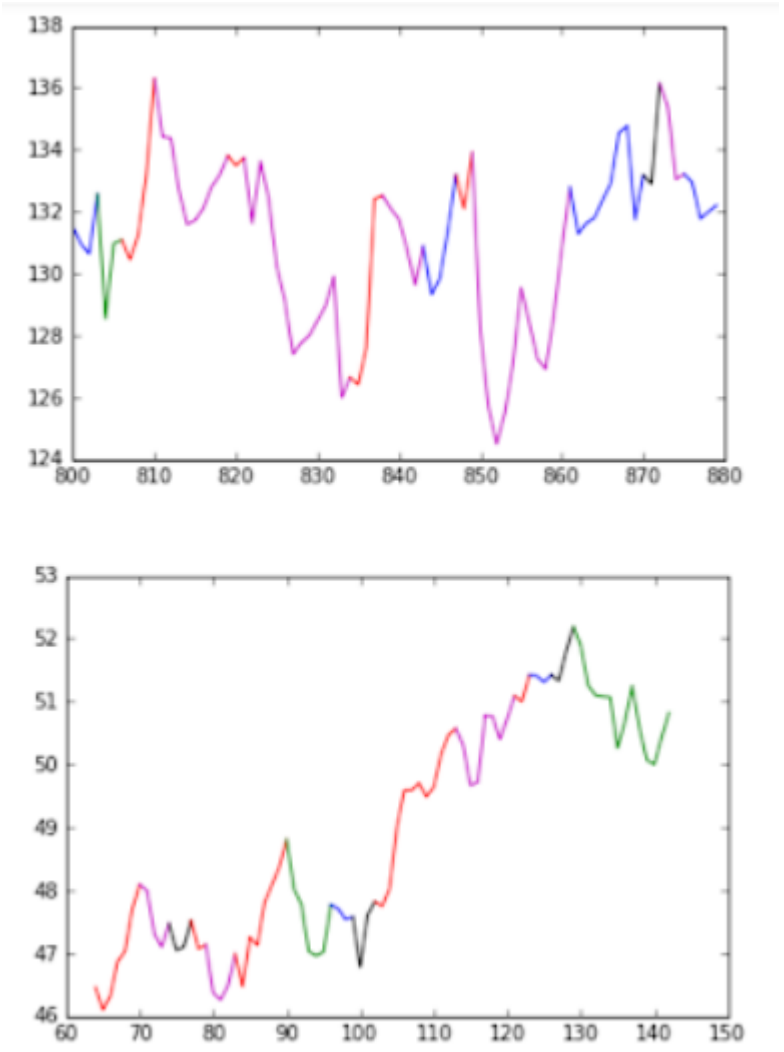

Figure 2: Examples of visualization color-coded by assigned groups

- On top of improving the vector grouping algorithm, we would also try to improve our predictive model. This may involve both increasing the order of the Markov model to use more context to make a prediction as well as changing the type of predictive models. Alternative models could use either unsupervised learning (e.g. support-vector machine or K-nearest neighbors) or supervised learning (e.g. multi-layered perceptrons.)

- After we select our models, we would also want to optimize the hyperparameters of our predictive model to achieve better performance.

Moreover, we plan to find more relevant information to aid the Markov model in making the prediction. One possible information is economic and political news which could affect the stock price of certain companies.

If time permits, we would also like to gain more insight into the stock data by developing an algorithm to extract combinatorial pattern in the data based on our grouping algorithms.

Acknowledgments

We would to express our appreciation to Professor Weiqing Gu for initiating our idea and advicing us throughout this project.

References

[1] Ou, Phichhang and Wang, Hengshan (2009) Prediction of Stock Market Index Movement by Ten Data Mining Techniques. *Modern Applied Science*, Vol. 3 No. 12.

[2] Saad, Emad W., Prokhorov, Danil V, Wunsch, Donald C. (1998) Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks. *IEEE Transactions on Neural Networks*, Vol. 9 No. 6.

[3] Tsai M-C, Cheng C-H, Tsai M-I, Shiu H-Y (2018) Forecasting leading industry stock prices based on a hybrid time-series forecast model. *PLoS ONE*.

[4] Zhang, Hui, Ho, Tu Bao, Zhang, Yang, Lin, Mao-Song (2006) Unsupervised Feature Extraction for Time Series Clustering Using Orthogonal Wavelet Transform. *Informatica*.

[5] Bonde, Ganesh and Khaled, Rasheed (2012) Extracting the best features for predicting stock prices using machine learning. *Proceedings of the 2012 International Conference on Artificial Intelligence*, ICAI 2012.