CSE 291, Winter 2017

Kuang Hsuan Lee
PID: A53204571
Email: khl085@eng.ucsd.edu

Po-Ya Hsu
PID: A53202971
Email: p8hsu@eng.ucsd.edu

I. Introduction

We are interested in making big money and are eager to explore how the prices in financial market flow with time. People desire to predict the market prices with historical information or fundamental information from news. However, both are difficult to predict market prices. With the maturity of artificial intelligence (AI) in recent years, we are equipped with some tools to make better and reasonable predictions. Thus, we are excited to implement some AI methods on stock market prices prediction.

Report Organization : First, the related work is briefly introduced in **Section II**. In **Section III**, we descibe the dataset used in this project. Then, we depict the models and conducted experiments in **Section IV** and **Section V** respectively. As for the models, three models are used in this project : traditional Support Vector Regression (SVR), Firefly SVR, and Wavelet Analysis SVR. Last, we make our conclusion in **Section VI** and place the referenced work in **Section VII**.

II. Related Work

A. Traditional Statistical Models

Everyone desires to find the pattern in stock market. However, stock market prediction is an extremely difficult task of financial time series prediction. The challenge of forecasting arises from the non-stationarity and non-linearity of the stock market and financial time series. In the past, researchers usually used the autoregressive (AR) model and the autoregressive integrated moving average (ARIMA) model to tackle this problem. However, these models were developed based on the assumption that the time series data are linear and stationary. After several years, researchers also proposed other nonlinear approaches, such as autoregressive conditional heteroscedasticity (ARCH) [1], generalized autoregressive conditional heteroscedasticity (GARCH) [2], artificial neural networks (ANNs) [3-8], fuzzy neural networks (FNN) [9-12], and support vector regression (SVR) [13-21].

B. Nonlinear Approaches: ANN and SVR

Last year, along with Alpha Go beats the Go champion, there are more and more researchers applied the ANN to any field they know. Financial market is also in the list. In fact, ANN provides the universal approximation property[22], which is good for modeling stock market time series. Previous researchers have indicated that ANN, which minimizes the empirical risk minimization principle in its learning process, outperforms traditional statistical models [3]. However, ANN also has its cons. It is difficult to determine the hidden layer size and learning rate and it is easily go into local minimum traps[23,24].

In contrary, support vector regression, originally proposed by Vapnik [23,25], has a global optimum and has better prediction accuracy since it implements the structural risk minimization principle, considering both the capacity of the regression model and the training error[24,26]. However, not every one can use SVR easily since it is difficult to determine its hyperparameters, which requires practitioner experience. The performance will be terrible if we choosing unsuitable kernel functions or hyperparameter settings[26-29].

C. Feature selection

It has been many years that people research what kind of features can predict the stock market. There are several usual features. First is the previous stock prices, inspired from similar with AR or ARIMA model. Second is the the volume since when the volume is high, it means there are more people trading this stock, which lead to the stock will increase or decrease more drastically.

III. Dataset

We crawled the stock market data from Yahoo Finance. The dataset is composed of stock prices of 2 companies (Intel and Microsoft), ranging from 1990/1/1 to 2016/12/31. Each company has 6805 data points, and each data point contains 6 information representing the trading information on that day: opening price, closing price, highest price, lowest price, volume, and adjusted closing price. The closing price is what we aim to predict and we focus on predicting the future adjusted closing price.

Therefore, we put the volume and the adjusted closing price into our model to transfer to new feature spaces and put it together to do prediction.

For each company, we use the first 80% data points for training, and the rest 20% for testing. That is, all the data points before 2011/08/04 are used for training, and the rest are used for testing.

IV. Model

A. Deciding the time delay and the dimension

Since the stock market is non-linear dynamic systems, which can be done in accordance with Takens embedding theory[30]. We can express the stock prices as the time series $\{X_i\}_1^N$, where N is the length of the time series. We would like to change it to the (N-m) m space. And the equation can be expressed as the following:

$$P_i = P_{i-\tau} + P_{i-2*\tau} + ... + P_{i-(m-1)*\tau} \quad (1)$$

where m is called the embedding dimension of reconstructed phase space and τ is the time delay constant.
Deciding the delay term τ is a trade-off. Because if τ is too large, it will probably cause an irrelevance phenomenon and if τ is too small, redundancy will occur. In this study, we use the first minimum of mutual information (MI) function [31] to determine τ as follows:

$$P_i = P_{i-\tau} + P_{i-2*\tau} + ... + P_{i-(m-1)*\tau} \quad (2)$$

And we can apply false nearest neighbors (FNN) procedure, proposed by Kennel et al.[32], to efficiently find the minimal sufficient embedding dimension. Two near points in reconstructed phase space are called false neighbors if they are significantly far apart in the original phase space. Such a situation occurs if we select an embedding dimension lower than the minimal sufficient value and therefore the reconstructed attractor does not preserve the topological properties of the real phase space. That is, points are projected into the false neighborhood of other points. The idea behind the FNN procedure is as follows Suppose X_i has a nearest neighbor X_j in an m-dimensional space. Calculate the Euclidean distance$||X_i X_j||$ and compute

$$\sum_{n=1}^{N-\tau} P(X_n, X_{n+\tau}) log_2 \frac{P(X_n, X_{n+\tau})}{P(X_n)P(X_{n+\tau})} \quad (3)$$

B. Chaos-based Firefly Algorithm for selecting parameters for SVR

After getting the reconstruct space, we begin to find the parameters for SVR with Chaos-based Firefly Algorithm[32]. The firefly algorithm(FA) is a meta-heuristic optimization algorithm inspired by the flashing behavior of fireflies[33]. In this algorithm, there are several fireflies, standing for one parameter. The fireflies with lower light, lower performance leading by this parameter, will be close to the fireflies with high light and the fireflies with highest light only moves randomly. However, random terms sometimes is not best choice, which not leading searching the whole solution space. Hence we used Chaos-based Firefly Algorithm (CFA) to search the whole solution space.

The chaotic firefly algorithm are as follows steps until converge:

Step 1: Generate initial positions of fireflies by chaotic mapping operator (CMO)[34]. The values of the three hyper-parameters are C, ϵ and γ. If we have 20 fireflies, that is, we have 20 parameters pairs. For each parameter, we transfer its value to the interval (0,1) with max and min and used logical mapping to generate the new value and transfer back to the actual value with max and min again. The logical mapping is the following formula:

$$x_{i+1} = 4 * x_i * (1 - X_i) \quad (4)$$

Step2: For each firefly, we compute its light intensity, the performance with mean absolute percentage error (MAPE) as the fitness function.

Step3: After knowing the light intensity of each firefly, they will move chaotically. For more precisely, the fireflies with lower performance move toward fireflies with higher performance. The firefly with the highest performance moves chaotically in the solution space to search the whole solution space.

The following formula is the CFA updating algorithm:

$$x_i = x_i + \beta(x_j - x_i) + 1 - \|\frac{n-1}{n}\|^v \quad (5)$$

$$\beta = \beta_0 * exp(-\lambda * r_{ij}^2) \quad (6)$$

where x_i is a firefly with higher light intensity, x_j is a firefly with lower light intensity; λ is the absorption coefficient, r_{ij} is the Euclidean distance between x_i and x_j; β_0 is the maximum attractiveness value and n is the iteration number and v is an integer.
After the movements, all fireflies move toward the neighborhood of the best firefly, improving their personal performance. The firefly with the highest light intensity moves Chaos to search the global solution. After reaching the iterations defined, the firefly with the highest performance is our solution.

Step4: Stopping condition. If the number of iterations is equal to the iteration defined, then the firefly with highest performance is regarded as a solution; otherwise go back to step 2.

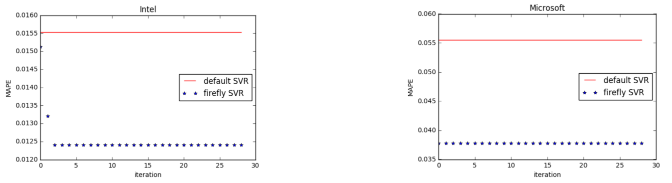
C. Wavelet Analysis to extract temporal information

Our hypothesis is that different eras have their unique fingerprints. Given the fact that the era's duration is unknown, we use maximal overlap discrete wavelet analysis (MODWT) and multiresolution analysis to extract the temporal variation features. [35] Then, we apply statistical analysis to compute the estimated changing points of the features, and claim that they approximately segment each market era. For each segmentation, we believe it represents the market trend in its time, and has completed experiments to test our hypothesis with fixed sizes of training and testing data under different circumstances.

V. Experiment

A. Experiment setup

As described in Sec.III, we use the first 80% of data for training, and the rest 20% for testing. Note that for the first



(a) Intel

(b) Microsoft

Fig. 1: The MAPE with iteration growing

few instances in the testing set, their features are generated by the last few instances in the training set, since some of our features are delayed.

To evaluate the performance, we adopt Mean Absolute Percentage Error (MAPE) as our evaluation metrics, which is defined as:

$$MAPE = \frac{1}{N}\sum_{i}^{N}\frac{|y_i - \hat{y}_i|}{y_i}, \tag{7}$$

where y is the actual value, and $\hat{y}$ is the predicted value. MAPE is commonly used to measure forecasting error, which suits our need very well.

We mainly perform two experiments. The first one is that we would evaluate the performance of SVR with firefly algorithm, as described in Sec.IV-B, and the second one is that we would segment the pattern to several slots and do prediction.

B. Firefly on SVR model

From the Figure 1, we found the firefly algorithm is stable as the iteration grows. For Intel, even if at first we would not find the best solution, we used chaotic behavior to let the firefly to search the whole space to find a better solution while other fireflies with lower performance moves to the firefly with high performance. That is, we do not give up the current best solution and also we can keep finding other better solution. For Microsoft, we find the best solution in the beginning and we let other fireflies to move to it. Hence, we still hold the current best solution.

The result is in the table I:

stock	default SVR	Firefly SVR
Intel	0.01552	0.01262
Microsoft	0.05553	0.03775

TABLE I: SVR Performance

From these two tables above, we can safely conclude that we almost can get better result with firefly algorithm than the default parameters in SVR.

C. Wavelet Analysis SVR Experiment

To testify the correctness of our hypothesis, we first find the changing points by MODWT and multiresolution analysis. Then, we conduct SVR on each era with fixed training and testing data size. Moreover, we perform SVR on shifted data, which gives rise to cross-era prediction results, and can therefore be compared to within-era results to verify our hypothesis.

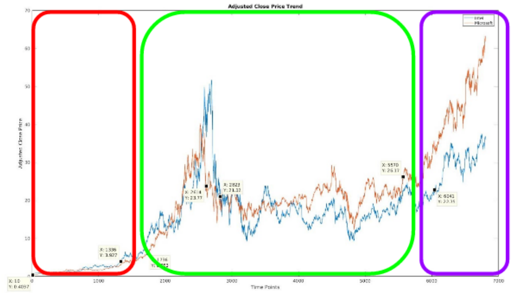


Fig. 2: Three Eras found in Intel and Microsoft Data

For both Intel and Microsoft data, two changing points are found, which is equivalent to three eras discovered and is clearly shown in Fig. 2. For SVR results, they are listed in TABLE II to TABLE VII. The standard within era MAPE results are in italic style, whereas for those meeting our hypothesis are in bold style.

Through the observation of the tables' MAPE results, we have confidence that the changing points are successfully found in Microsoft stock market, but not in Intel.

shift	training	testing	MAPE
0%	1-1350	1351-1736	*0.1355*
50%	194-1543	1544-1929	0.1897
100%	387-1736	1737-2122	**0.3755**

TABLE II: Intel Era A : SVR Performance

shift	training	testing	MAPE
0%	1-1050	1051-1336	*0.1906*
50%	144-1193	1194-1479	0.1818
100%	287-1336	1337-1622	**0.3372**

TABLE III: Microsoft Era A : SVR Performance

shift	training	testing	MAPE
0%	1337-5100	5101-6041	*0.084*
50%	1807-5570	5571-6511	0.0409
100%	2247-6040	6041-6805	0.0134

TABLE IV: Intel Era B : SVR Performance

shift	training	testing	MAPE
0%	1737-4950	4951-5569	*0.0322*
50%	2047-5260	5261-5880	**0.0481**
100%	2357-5570	5571-6190	**0.0667**

TABLE V: Microsoft Era B : SVR Performance

shift	training	testing	MAPE
0%	6041-6680	6681-6805	*0.0525*
50%	5722-6361	6362-6485	**0.1106**
100%	5402-6041	6042-6165	0.0203
overall%	1-6680	6681-6805	0.0124

TABLE VI: Intel Era C : SVR Performance

shift	training	testing	MAPE
0%	5570-6500	6501-6805	*0.0205*
50%	5106-6035	6036-6335	**0.2144**
100%	4641-5570	5571-5875	**0.24666**
100%	1-6500	6501-6805	**0.0220**

TABLE VII: Microsoft Era C : SVR Performance

VI. Conclusion

From the Firefly SVR experiments, we can conclude that our firefly algorithm is useful for selecting SVR parameters, which introduced from the data set or the features. For other problems, the firefly algorithm is also a good method to find the parameters, not limited to stock price prediction.

Regarding the Wavelet Analysis SVR experiment, we are unable to claim if different trends occur in their belonging eras. We successfully captured Microsoft stock market trend in each era, but not Intel. There are two possible reasons. One is the mathematical analysis itself should be improved to find appropriate changing points. The other is the fact that the changing points do not exist in Intel stock market. Since we have tried on only two data, we conclude more tests should be completed to state the existence of eras in stock markets.

A. future work

We need more data and higher resolution of market data. If we have these data, we not only compare the performance of segmentation easily but also generate the prediction model with financial news text mining since when the news publish, which affecting the stock price in few minutes and it would easily clarify whether the news is positive or negative. With model from text mining, we can build a model from outside information and merge it to our current model, using the historical data and the transparent of the information of the company to predict the future prices. We expect that if the company is more transparent, the model from text mining is more predictive.

References

[1] R.F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation, Econometrica 50 (1982) 9871008.

[2] T. Bollerslev, Generalized autoregressive conditional heteroscedasticity, Journal of Econometrics 31 (1986) 307327.

[3] J.V. Hansen, R.D. Nelson, Neural networks and traditional time series methods: a synergic combination in state economic forecasts, IEEE Transactions on Neural Networks 8 (1997) 863873.

[4] K.J. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Systems with Application 19 (2008) 125132.

[5] Y.K. Kwon, B.R. Moon, A hybrid neurogenetic approach for stock forecasting, IEEE Transactions on Neural Networks 18 (2007) 851864.

[6] M. Qui, G.P. Zhang, Trend time series modeling and forecasting with neural networks, IEEE Transactions on Neural Networks 19 (2008) 808816.

[7] D. Zhang, L. Zhou, Discovering golden nuggets: data mining in financial applications, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34 (2004) 513522.

[8] L. Yu, S. Wang, K.K. Lai, A neural-network-based nonlinear meta modeling approach to financial time series forecasting, Applied Soft Computing 9 (2009) 563574.

[9] P.C. Chang, C.H. Liu, A TSK type fuzzy rule based system for stock price prediction, Expert Systems with Applications 34 (2008) 135144.

[10] S.K. Oh, W. Pedrycz, H.S. Park, Genetically optimized fuzzy polynomial neural networks, IEEE Transactions on Fuzzy Systems 14 (2006) 125144.

[11] M.H.F. Zarandi, B. Rezaee, I.B. Turksen, E. Neshat, A type-2 fuzzy rule based expert system model for stock price analysis, Expert Systems with Applications 36 (2009) 139154.

[12] C.F. Liu, C.Y. Yeh, S.J. Lee, Application of type-2 neuro-fuzzy modeling in stock price prediction, Applied Soft Computing 12 (2012) 13481358.

[13] L. Cao, F.E.H. Tay, Financial forecasting using support vector machines, Neural Computing and Applications 10 (2001) 184192.

[14] L. Cao, F.E.H. Tay, Support vector machine with adaptive parameters in financial time series forecasting, IEEE Transactions on Neural Networks 14 (2003) 15061518.

[15] P.C. Fernando, A.A.R. Julio, G. Javier, Estimating GARCH models using support vector machines, Quantitative Finance 3 (2003) 163172.

[16] T.V. Gestel, J.A.K. Suykens, D.E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, Financial time series prediction using least squares support vector machines within the evidence framework, IEEE Transactions on Neural Networks 12 (2001) 809821.

[17] P.F. Pai, C.S. Lin, A hybrid ARIMA and support vector machines model in stock price forecasting, Omega: The International Journal of Management Science 33 (2005) 497505.

[18] F.E.H. Tay, L. Cao, Application of support vector machines in financial time series forecasting, Omega: The International Journal of Management Science 29 (2001) 309317. G. Valeriy, B. Supriya, Support vector machine as an efficient framework for stock market volatility forecasting, Computational Management Science 3 (2006) 147160.

[19] H. Yang, L. Chan, I. King, Support vector machine regression for volatile stock market prediction, in: Proceedings of The Third International Conference on Intelligent Data Engineering and Automated Learning, 2002, pp. 391396.

[20] K.J. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (2003) 307319.

[21] V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models, MIT Press, Cambridge, MA, 2001.

[22] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[23] C.Y. Yeh, C.W. Huang, S.J. Lee, A multiple-kernel support vector regression approach for stock market price forecasting, Expert Systems with Applications 38 (2010) 21772186.

[24] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, Cambridge, UK, 2000.

[25] C.Y. Yeh, C.W. Huang, S.J. Lee, A multiple-kernel support vector regression approach for stock market price forecasting, Expert Systems with Applications 38 (2011) 21772186.

[26] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Machine Learning 46 (2002) 131159.

[27] K. Duan, S. Keerthi, A.N. Poo, Evaluation of simple performance measures for tuning SVM hyperparameters, Neurocomputing 51 (2003) 4159.

[28] J.T.Y. Kwok, The evidence framework applied to support vector machines, IEEE Transactions on Neural Networks 11 (2000) 11621173.

[29] F. Takens, Detecting strange attractors in turbulence, Lecture Notes in Mathematics 898 (1981) 366381.

[30] M. Kennel, R. Brown, H.D.I. Abarbanel, Determining embedding dimension for phase space reconstruction using geometrical construction, Physical Reviews A 45 (1992) 34033411.

[31] M. Kennel, R. Brown, H.D.I. Abarbanel, Determining embedding dimension for phase space reconstruction using geometrical construction, Physical Reviews A 45 (1992) 34033411.

[32] Ahmad Kazema, Ebrahim Sharifia, Farookh Khadeer Hussain, Support vector regression with chaos-based firefly algorithm for stock market price forecasting

[33] X.S. Yang, Nature Inspired Metaheuristic Algorithms, Luniver Press, Frome, UK, 2008.

[34] W.C. Hong, Y. Dong, L.Y. Chen, S.Y. Wei, SVR with hybrid chaotic genetic algorithms for tourism demand forecasting, Applied Soft Computing 11 (2010) 18811890.

[35] Alarcon-Aquino, V., and J. A. Barria. "Change detection in time series using the maximal overlap discrete wavelet transform." Latin American applied research 39.2 (2009): 145-152.