Stock Price Prediction

Kuang Hsuan Lee
Po-Ya Hsu
CSE Dept. UC San Diego

Outline

- Motivation
- Related Work
- Our Data & Model
- Experiment
- Results
- Conclusion
- References

Motivation

1. People desire to predict the market prices with historical information.
2. However, it is very difficult to predict market prices because there are three challenging issues in financial time series processing are noise, non-linearity and non-stationarity.
3. With the maturity of artificial intelligence (AI) in recent years, we are equipped with some tools to make better and reasonable predictions.
4. We are interested in making big money and are eager to explore how the prices in financial market flow with time

Related Work (Traditional Statistics)

1. Traditional methods such as Autoregressive (AR) model
2. autoregressive integrated moving average (ARIMA) model
3. Both are linear and stationary time series.

.

Related Work (ANN)

1. artificial neural networks (ANN) which minimize empirical risk principle in its learning process, can handle this non-linear problem

2. It also has problems about local minimum traps and the difficulty to decide hidden layer size and learning rate.

Related Work (Support Vector Regression)

1. SVR is better than ANN because it uses structural risk minimization principle which considers both the training error and the capacity of the regression model to minimize the upper bound of generalization error.

2. The main problem with SVR is that we require practitioner experience to determine its hyperparameters and kernel functions.

3. Also, the stock market is not constant pattern, we should segment the pattern and do prediction

Our Data & Model

Our Data

1. We obtain Intel and Microsoft data from Finance Yahoo from 1990/1/1 to 2016/12/31

2. We focus on predicting the adjusted close price

3. We used stock price and volume to do prediction

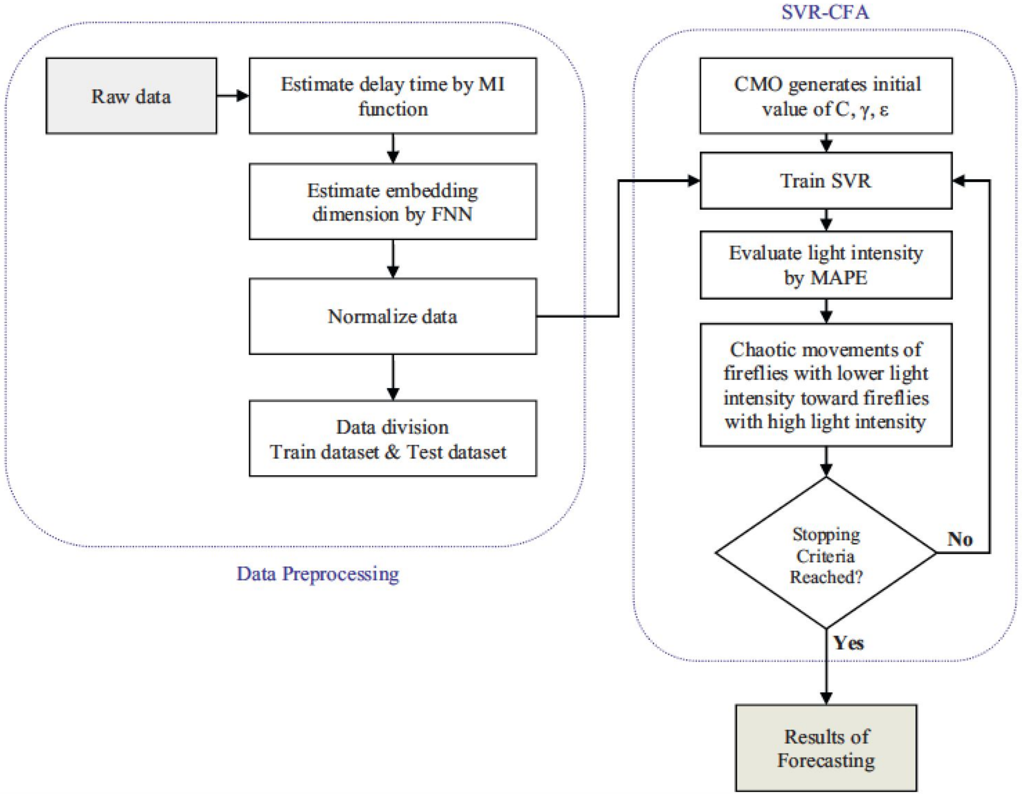
Our Model

1. For adjusted stock price and volume to reconstruct the feature space with delay term and dimension.

2. For the trend of these stocks to do segmentation.

3. We used chaotic firefly algorithm for optimizing the SVR hyperparameters is good for finding hyperparameters for SVR

Our Model

1. 

SVR-CFA
Raw data
Estimate delay time by MI function
Estimate embedding dimension by FNN
Normalize data
Data division
Train dataset & Test dataset
Data Preprocessing
CMO generates initial value of C, γ, ε
Train SVR
Evaluate light intensity by MAPE
Chaotic movements of fireflies with lower light intensity toward fireflies with high light intensity
Stopping Criteria Reached?
No
Yes
Results of Forecasting

Our Model - deciding delay term and dimension

1. Reconstruct the feature space with m and tau

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

2. We use the first minimum of mutual information (MI) function to determine

$$\mathrm{MI}\,(\tau) = \sum_{n=1}^{N-\tau} P(x_n, x_{n+\tau}) \log_2\left(\frac{P(x_n, x_{n+\tau})}{P(x_n)P(x_{n+\tau})}\right)$$

Our Model - deciding delay term and dimension

1. Reconstruct the feature space with m and tau

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

2. We use the first minimum of mutual information (MI) function to determine

$$\mathrm{MI}\ (\tau) = \sum_{n=1}^{N-\tau} P(x_n, x_{n+\tau}) \log_2 \left(\frac{P(x_n, x_{n+\tau})}{P(x_n) P(x_{n+\tau})} \right)$$

Our Model - deciding delay term and dimension

1. Using false nearest neighbors: Suppose Xi has a nearest, neighbor Xj in an m-dimensional space. Calculate the Euclidean distance ||Xi −Xj|| and compute Ri, if Ri > 10, it is false nearest neighbors.

$$R_i = \frac{||X_{i+1} - X_{j+1}||}{||X_i - X_j||}$$

2. the point Xj is considered as a false nearest neighbor in dimension m.
3. We can say that the embedding dimension m is sufficiently high if the fraction of points that have false nearest neighbors is zero or considerably small

Our Model - Logical mapping

1. Generate the chaotic behaviour can arise from very simple non-linear dynamical equations

$$x_p^{(i)} = \frac{X_p^{(i)} - Min_p}{Max_p - Min_p}, \quad p = C, \gamma, \varepsilon$$

$$x_{n+1} = \mu x_n(1 - x_n)$$

$$X_p^{(i+1)} = Min_p + x_p^{(i+1)}(Max_p - Min_p)$$

Our Model - FireFly algorithm

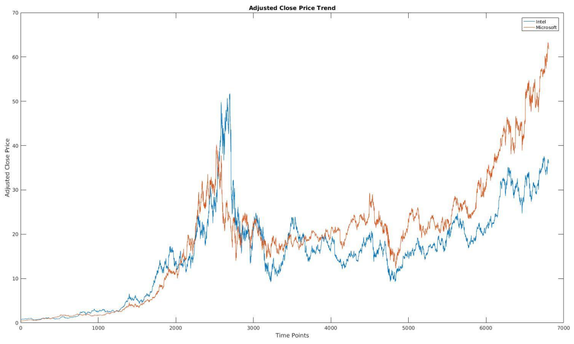
1. The fireflies with lower performance move toward fireflies with higher performance. The firefly with the highest performance moves chaotically in the solution space to search the whole solution space.

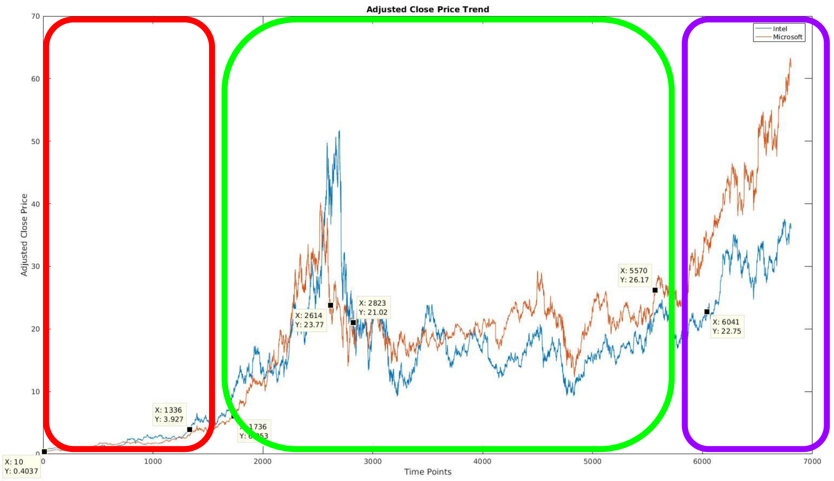
$$x_i = x_i + \beta(x_j - x_i) + 1 - \|\frac{n-1}{n}\|^v$$

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

Our Model - trend segmentation

1. Apply maximal overlap discrete wavelet analysis and multiresolution analysis to extract the temporal variation features from our data
2. Find the possible segmentation of our data via the computation of estimated variance change points from the features captured in Step 1
3. Pick the significant segmentation points

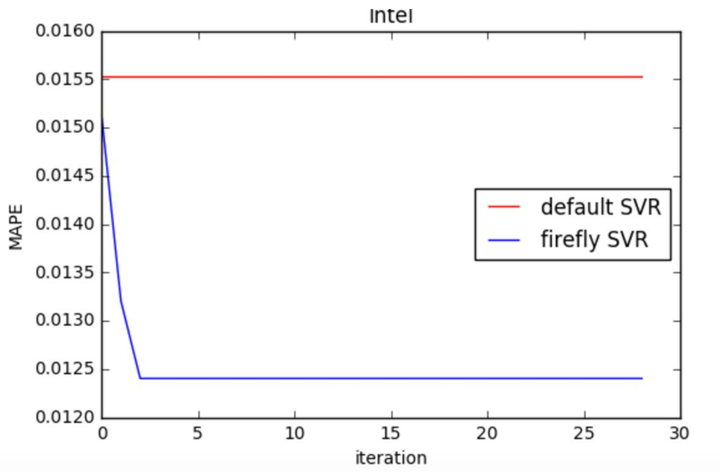


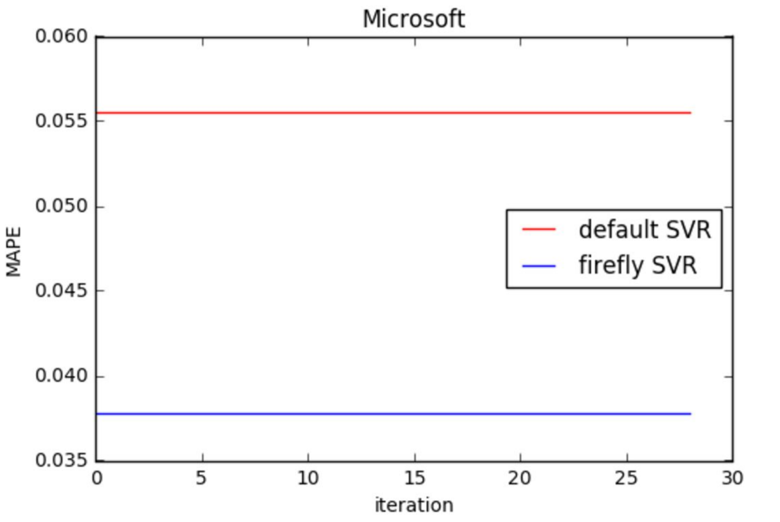


Experiment 1

FireFly algorithm v.s. Default SVR

1.





FireFly algorithm v.s. Default SVR

1.

	default_SVR	firefly_SVR
Intel	0.01552	0.01262
MicroSoft	0.05553	0.03775

Experiment 2

Our Hypothesis

Different eras have their own stock market trends / patterns

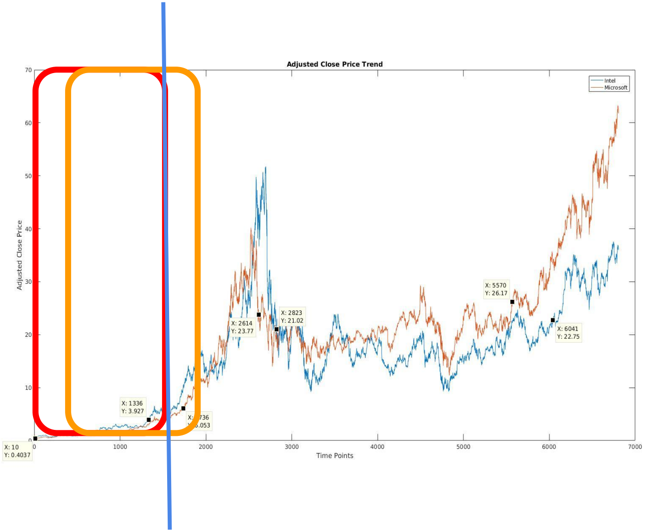
Procedure

For each pattern experiment,

- Run SVR with fixed training and testing size

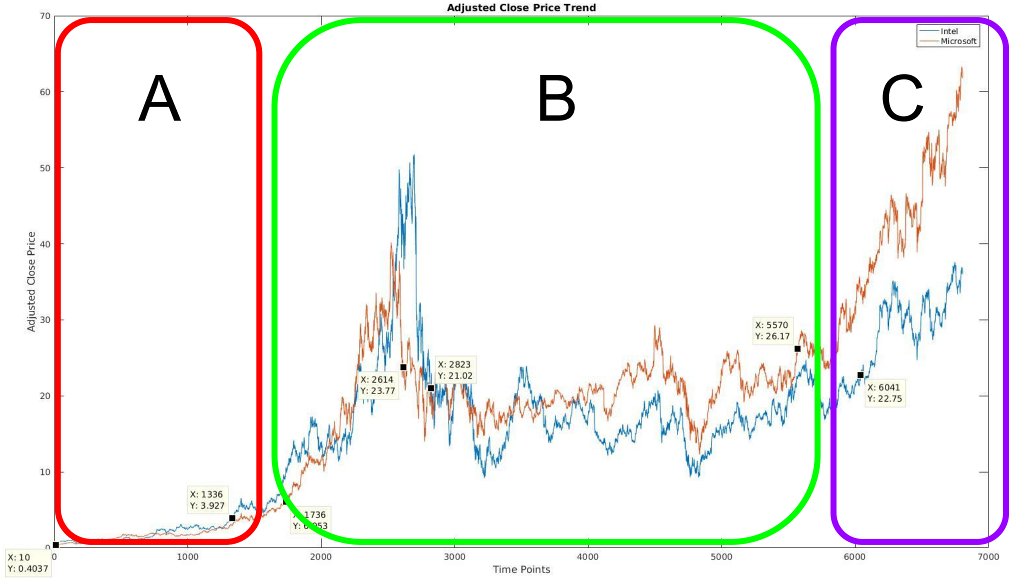
For each segmentation,

- Try the within, cross, and beyond pattern tests

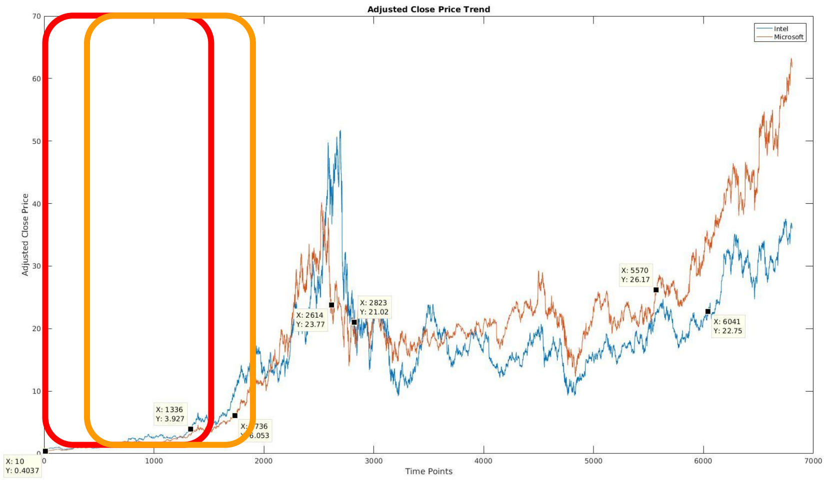


Result

Patterns : A, B, C



Segmentation SVR



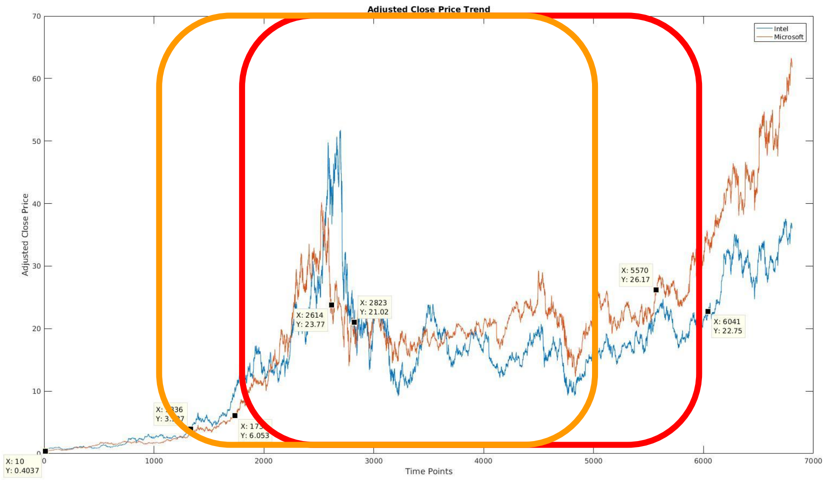
Intel

shift		training	testing	MAPE
	1	1350	1736	0.1355
50%	194	1543	1929	0.1897
100%	387	1736	2122	0.3735

Microsoft

shift		training	testing	MAPE
	1	1050	1336	0.1906
50%	144	1193	1479	0.1818
100%	287	1336	1622	0.3372

Segmentation SVR



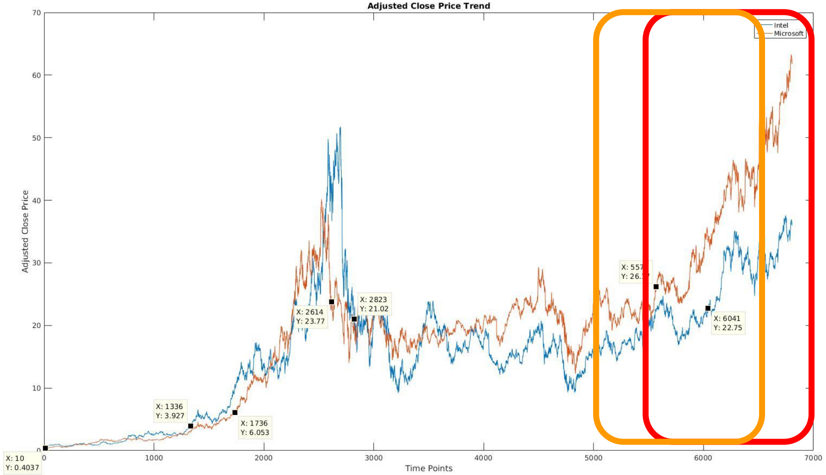
Intel

shift	training		testing	MAPE
	1337	5100	6041	0.084
50%	1807	5570	5571	0.0409
100%	2247	6040	6041	0.0134

Microsoft

shift	training		testing	MAPE
	1737	4950	5569	0.0322
50%	2047	5260	5880	0.0481
100%	2357	5570	6190	0.0667

Segmentation SVR



Intel

shift	training		testing	MAPE
	6041	6680	end	0.0525
50%	5,722	6,361	6485	0.1106
100%	5,402	6,041	6165	0.0203
overall	1	6,680	end	0.0124

Microsoft

shift	training		testing	MAPE
	5570	6500	end	0.0205
50%	5106	6035	6335	0.2144
100%	4641	5570	5875	0.2466
overall	1	6500	end	0.022

Conclusion

1. Firefly Algorithm works and makes improvement on SVR

2. The patterns in stock market probably exist. In our experiments, we are able to identify Microsoft's patterns, but not Intel's. Perhaps a better segmentation approach should be implemented to trace the patterns out.

Next Step

1. Acquire larger data size with higher resolution
2. In such data scale, we can have more data wtih the fixed segmentation size.
3. A better segmentation approach should be implemented to trace the patterns out.
4. With higher resolution, we can build the other fundamental based prediction model with the features from financial news text mining and merge our current mode: historical market information as the integrated model to tackle the stock prediction with different information efficiency.

References

1. Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, *29*(4), 309-317.
2. Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on* (Vol. 6, pp. 348-353). IEEE.
3. Oldewurtel, F., Parisio, A., Jones, C. N., Gyalistras, D., Gwerder, M., Stauch, V., ... & Morari, M. (2012). Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, *45*, 15-27.
4. Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine*, *4*(2).
5. Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K. (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied soft computing*, *13*(2), 947-958.
6. Choudhury, S., Ghosh, S., Bhattacharya, A., Fernandes, K. J., & Tiwari, M. K. (2014). A real time clustering and SVM based price-volatility prediction for optimal trading strategy. *Neurocomputing*, *131*, 419-426.
7. Hsu, S. H., Hsieh, J. P. A., Chih, T. C., & Hsu, K. C. (2009). A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Systems with Applications*, *36*(4), 7947-7951.
8. Xiaodong Li • Haoran Xie • Ran Wang(2016). Empirical analysis: stock market prediction via extreme learning machine
9. 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.