A Relative Tendency Based Stock Market Prediction System

ManChon U
Department of Computer Science
The University of Georgia
Athens, GA 30602, USA
manchonu@cs.uga.edu

Khaled Rasheed
Department of Computer Science & Institute for AI
The University of Georgia
Athens, GA 30602, USA
khaled@uga.edu

***Abstract*— Researchers have known for some time that non-linearity exists in the financial markets and that neural networks can be used to forecast market returns. In this article, we present a novel stock market prediction system which focuses on forecasting the relative tendency growth between different stocks and indices rather than purely predicting their values. This research utilizes artificial neural network models for estimation. The results are examined for their ability to provide an effective forecast of future values. Certain techniques, such as sliding windows and chaos theory, are employed for data preparation and pre-processing. Our system successfully predicted the relative tendency growth of different stocks with up to 99.01% accuracy.**

Keywords- Algorithms, Economics, Time Series Analysis, Neural Networks, Forecasting

I. INTRODUCTION

Important changes have taken place over the last two decades within the financial markets, including the use of powerful communication and trading platforms that have increased the number of investors entering the markets. Traditional capital market theory has also changed, and methods of financial analysis have improved. Stock-return forecasting has attracted the attention of researchers for many years and typically involves an assumption that fundamental information publicly available in the past has some predictive relationships to future returns for individual stocks or indices. Examples of such information include economic variables, exchange rates, industry - and sector - specific information, and individual corporate financial statements. This is contrary to the general ideal of the efficient market hypothesis which states that all available information affecting the current stock value is absorbed by the market before the general public can make trades based on it. Therefore, some people believe that it is impossible or futile to forecast future returns as current prices already reflect all information currently known about the stocks.

This is still a debatable issue since there is contradictory evidence that markets are not fully efficient, and that it is possible to predict the future returns with results that are better than random by means of publicly available information such as time-series data on financial and economic variables. These studies indicate that variables such as interest rates, monetary-growth rates, changes in industrial production, and inflation rates are statistically important for predicting a portion of the stock returns. However, most of the studies that attempt to capture the relationship between the available information and the stock returns rely on simple linear regression assumptions, even though there is no evidence that the relationship between the stock returns and the financial and economic variables is linear. Since there is significant residual variance of the actual stock return from the prediction of the regression equation, it is possible that nonlinear models could be used to explain this residual variance and produce more reliable predictions of the stock price movements.

Neural network learning is a nonlinear modeling technique that may overcome these problems. Neural networks offer a novel technique that does not require pre-specification during the modeling process since they independently learn the relationship inherent in the variables. This is especially useful in securities investment and other financial areas where much is assumed and little is known about the nature of the processes determining asset prices. Neural networks also offer the flexibility of various architecture types and learning algorithms.

The rest of this article is organized as follows: Section II describes the background of this research work. Section III presents the design of our prediction system. Section IV briefly discusses the types of people who would benefit from our system, and why. The experimental results are presented in section V. Finally, we present the conclusion in Section VI.

II. BACKGROUND

A. Stock Market Index

A stock market index is a list of stocks and a statistic reflecting the composite value of its components. There are many major market indices in the finance world; each of them tracks the performance of a specific group of stocks considered to represent a particular market or sector of the stock market or the economy. In this article, we use the Dow Jones Industrial Average (DJIA) as reference. DJIA is an index of the stocks of 30 of the largest and most widely held public companies in the United States. The Index includes substantial industrial companies with a history of successful growth and wide investor interest. The Index includes a wide range of companies from financial services companies, to computer companies, to retail companies. Like most data in economics and finance, a stock market index has the form of a time series exhibiting very high noise, and significant non-stationary behavior and nonlinearity. Every one of those aspects will be studied later in this paper and techniques will be employed to overcome those difficulties.

B. *Efficient Market Hypothesis*

The efficient market hypothesis (EMH) was initially proposed by Professor Eugene Fama at the University of Chicago in the early 1960s [1][2][3][4], it is one of the most controversial and influential arguments in the field of academic finance research and financial market prediction. The crux of the EMH is that it should be impossible to predict trends or prices in the market through fundamental analysis or technical analysis. EMH is usually tested in the form of a random walk model. The random walk hypothesis was first proposed by Louis Bachelier [5]. A random walk is the path of a variable over time that exhibits no predictable patterns. If a price time series y(t) moves in a random walk, the value of y(t) in any period will be equal to the value of the previous period y(t-1), plus some random variable. The random walk hypothesis states that successive price changes between any two time periods are independent, with zero mean and variance proportional to the time interval between the two. In this case, the best indicator for future market prices is the present price.

Many early works [1][6][7][8] support the presentation of random walk model in the financial market. Yet, more and more recent studies [9][10][11][12] on financial markets argue that the market prices do not follow a random walk hypothesis. Whether the stock market index is predictable has been a continuous argument for a long time.

C. *Financial Market Prediction Methods*

Nowadays, there are three mainstream methods regarding stock market predictions: Technical Analysis, Fundamental Analysis, and Machine Learning Algorithms.

Technical analysis is a security analysis technique that claims the ability to forecast the future direction of prices through the study of past market data, primarily price and volume. [13] There are three key assumptions in technical analysis: 1. Market action (price, volume) reflects everything. 2. Prices move in trends. 3. History repeats itself. Fundamental analysis is a security valuation method that uses financial and economic analysis to predict the movement of security prices. Fundamental analysts examine a company's financial conditions, operations, and/or macroeconomic indicators to derive the intrinsic value of its common stock [14]. Because of the large exchange volume of the stock market, more and more people believe that machine learning is an approach that can scientifically forecast the tendency of the stock market. Machine learning refers to a system capable of the autonomous acquisition and integration of knowledge. Our prediction task is mainly based on artificial neural networks (ANNs) algorithm as it has been proven repeatedly that ANNs can predict the stock market with high accuracy. ANNs are non-linear statistical data modeling tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

III. System Details

A. *System Architecture*

Figure 1. illustrates the architecture of our financial market prediction system. Given the historical finical market data of different stocks, how can we predict the future relative tendency among them? The first thing we need to do is data preprocessing, which is to transform all of those raw financial data into appropriate sequential data format. Then our system transforms the formatted data into suitable training patterns. Next, by feeding the training patterns into our prediction blackbox, the neural network starts to perform training and prediction, followed by certain post-processing on the preliminary prediction result from the neural network. Afterwards, the final prediction are generated and presented to the users. The following sections will introduce more detail information of every processing component inside our system.

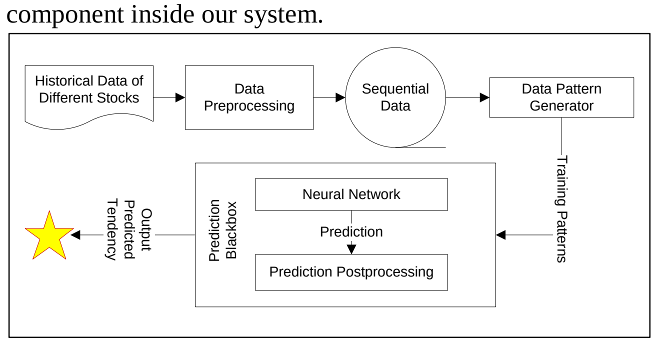

Figure 1. System Architecture

B. *Data Selection*

Data selection is essential in financial market prediction. The Dow Jones Industrial Average Index (DJIA) has a long history and it is widely referred to and studied. Numerous papers have discussed forecasting the tendency of DJIA, yet to the best of our knowledge, no research work has been done in this field for predicting the relative tendency growth between the DJIA and its component stocks. In this paper, we examine the DJIA and randomly select five of its component stocks from different areas as our reference data sample for the tendency prediction study, as shown in TABLE I.

TABLE I. INFORMATION OF STOCKS THAT WE STUDY IN THIS PAPER

Symbol	Stock Name	Passed 52wk Range
DJI	Dow Jones Industrial Average	8,057.57 - 11,309.00
T	AT&T Inc.	23.19 - 28.73
CAT	Caterpillar, Inc. Common Stock	30.01 - 72.83
INTC	Intel Corporation	15.78 - 24.37
JPM	JP Morgan Chase & Co. Common St	31.59 - 48.20
MMM	3M Company Common Stock	57.81 - 90.52

Furthermore, we have selected a recent period of 1024 days (4 years) from 02-Feb-2004 to 26-Feb-2008 as our study sample. We chose a length of 1024 days for two reasons. One reason is that, according to Peters [21], the Dow Jones Index has a four-year cycle. The other reason is that it provides enough data for a test, yet it does not suffer much from the Time Series Recency Effect [22]. Time Recency Effect is the phenomenon in building time series models that using data in time closer to the forecasted data produces better models.

C. Data Pre-Processing & Pattern Generator

In the financial domain, it is common to use log difference as daily return. Therefore, we firstly transform the historical data of different stocks into an appropriate sequential format, and then rescale the daily price indexes with logarithms. Afterwards, with the continuous 1024 trading days' time series data, constructing our training model becomes a critical point of the composition of our system. The sliding window technique is a popular approach to deal with time series data, in where N inputs and a single output slide over the whole time series. For example, given a time series $x_1, x_2, \ldots, x_{t-1}, x_t$, input vectors: $X_1 = (x_1, x_2, x_3, \ldots, x_N)$, $X_2 = (x_2, x_3, x_4, \ldots, x_{N+1})$, …, $X_i = (x_i, x_{i+1}, x_{i+2}, \ldots, x_t)$ are generated. Each input vector and a single output target value constitute one training sample for machine learning. Moreover, based on Taken's theorem [23], we know that given a time series $x_1, x_2, \ldots, x_i, x_{i+1}$, we can construct the underlying dynamical system by time-delay embedding vectors $X_i = (x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+(d-1)\tau})$ if we use appropriate d and τ values, where d is called the embedding dimension and τ is called the separation. Qian and Rasheed [19] discussed the details of how to obtain the popper d and τ. Based on their work, we decided to set the embedding dimension of our data sets to 1 (d=1). They also suggest that the separation should be from 3 to 5, and therefore throughout the course of the experiments, we set the separation to 4 (τ=4) for our system.

D. Neural Network Structure

There is no solid evidence to prove that neural networks with multiple hidden layers outperform those with a single hidden layer for continuous function prediction. Also the more complicated a neural network, the longer time it will require for both training and testing. Therefore, we have implemented our neural network with a single hidden layer structure.

Among the most popular practical algorithms for training artificial neural networks are Levengerg-Margqardt (LM) and backpropagation (BP) with momentum [17]. Throughout the course of our experiments, we found that the LM learning algorithm beats BP consistently on our data set. Thus we select LM algorithm with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. We then have to decide the number of hidden nodes. Based on the heuristic rule for determining the number of hidden nodes described in [15], the total degrees of freedom of a network should equal one and a half times the square root of the size of the data set. Thus we have Equation (3) below.

$$\begin{cases} f_1 = (\#inputNodes + 1) \times (\#hiddenNodes) \\ f_2 = (\#hiddenNodes + 1) \times (\#outputNodes) \\ f_3 = 1.5\sqrt{(\#datasetSize)} \\ f_1 + f_2 = f_3 \end{cases} \tag{1}$$

In our system, the size of the whole dataset is 1024 trading days, so the #datasetSize in Eq. (3) is 1024, and the number of input nodes and output nodes are 4 (dimension 4) and 1 respectively, which are introduced in section III.C above. Thus the solution for the number of hidden nodes for our system is 8 in this case.

As we mentioned before, the target of our system is not just predicting the price of certain stocks, but to go further and say, our system is on target to forecast the relative tendency growth among different stocks. Therefore, in this article, we select Dow Jones Industrial Average (DJIA) as a reference base, and try to forecast the relative performance between DJIA and five of its component stocks picked at random. In other words we predict whether a stock's relative performance would be better or worse than the DJIA. The data structure of the neural network processing unit in our system is, the input and target vector for training are a 4 x 820 matrix and a 1 x 820 vector, while the input and target vector for testing are a 4 x 204 matrix and 1 x 204 vector. This means that in our system, we separate our data set into two parts in order to avoid the over-fitting problem. The first 80% of the data (820 trading days) are utilized for training and validation, while the last 20% (204 trading days) are used as out-of-sample data for testing; all data within this 20% are never used in any model generation or validation process.

We use four continuous daily price indexes (4 dimension input value) of stock X to predict its price index in the fifth day (1 dimension target value). Our system first trains the neural network models of DJIA and its other five randomly picked stocks separately. This means that we will have six independent neural network models once the training process is finished; we then utilize these six individual models respectively for their corresponding stock prediction.

E. *Prediction Postprocessing*

After getting the prediction results from the neural network, the next step for handling these results in order to obtain highly reliable and accurate final prediction result becomes an issue. Since our goal is to tell a user which stock is going to perform better than the other, the most common way is to normalize the prediction of different stocks and put them into the same scale for comparison. DJIA will be announced to perform better (+ve) in the next day than the compared stock X if the normalized price index of DJIA in the next day is greater than the normalized price index of stock X and vice versa.

IV. BENEFICIARIES

Our system can predict the future relative tendency growth among different stocks – or stocks and indices - on a daily bases. Therefore, daily or short term investors and mutual fund managers – especially hedge fund managers - are the most likely group to benefit from our system as they can decide to perform the drop/add of a specific stock on a specific day.

A mutual fund can be viewed as a cooperative of investors. A number of people pool their investment money together to purchase stocks, bonds, securities of other investment tools. This fund is then managed by a fund manager who is presumably an expert in this area. Participants in the stock market can be based anywhere. They range from small individual stock investors to large mutual fund traders. Since the stock prices fluctuate widely, more and more people nowadays tend to entrust their money to the care of professional stock traders. Furthermore, the majority of the investors in the stock market tend to be “risk-avoiding”, which means that they prefer to have steadily small amounts of profit then to make big profits at high risk. Therefore, mutual funds became very popular in the last decade, as they are considered “safer” for individual investors in terms of getting returns.

V. EXPERIMENTAL RESULTS

A. *System Environment*

Our system is implemented using MATLAB, and we have constructed our ANN with the MATLAB Neural Network Toolbox. For ease of reference, the configuration of our ANN is summarized in Table II. Each network model is trained for 300 cycles (epochs) in maximum, while the MSE (Mean Square Error) is being calculated and recorded. The definition of MSE is described in Eq. (4), in where O is the output value and T is the target value.

TABLE II. CONFIGURATION OF THE ANN IN THIS PAPER

Parameter		Value		
# of hidden layer(s)		1		
# of node(s) in each layer	Layer	Input	Hidden	Output
	Value	4	8	1
Transfer functions		tansig	tansig	purelin
Learning Rule		Levengerg-Margqardt		
Epochs		300		
Error limit		1e-10		
Data set size (Trading days)		Training + Validation: 820		
		Testing: 204		

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(O_j - T_j)^2 \quad (2)$$

B. *Result Analysis*

TABLE III. MSE & CORRESPOINDING STANDARD DEVIATION OF THE NEURAL NETWORK MODEL OFTRAINING DATA SET

Run	DJI	ATT	CAT	INTC	JPM	MMM
1	0.5797	3.0181	16.8693	4.3862	1.7968	2.2876
2	0.5916	1.9264	17.3327	4.2236	1.7392	2.2736
3	0.5891	1.8354	14.4925	4.1357	1.7622	2.2375
4	0.6834	1.8344	14.4137	4.1907	1.7692	2.2808
5	0.5891	1.8707	14.6411	4.1699	1.7679	2.4157
6	0.5907	1.8937	14.8198	4.1737	1.7899	2.2766
7	0.5871	1.9513	14.9961	4.1461	1.7480	2.2570
8	0.5885	1.9616	14.9956	4.1022	1.7180	2.2525
9	0.5883	2.0797	15.0241	4.1287	1.7729	2.2570
10	0.6192	2.0229	14.9392	4.3051	1.7629	2.2863
Avg.	**0.6007**	**2.0394**	**15.2524**	**4.1962**	**1.7627**	**2.2825**
Std.	**0.0308**	**0.3527**	**1.0036**	**0.0878**	**0.0231**	**0.0495**

a. Vales are on the scale of $1e^{-5}$

TABLE IV. MSE & CORRESPOINDING STANDARD DEVIATION OF THE NEURAL NETWORK MODEL OF TESTING DATA SET

Run	DJI	ATT	CAT	INTC	JPM	MMM
1	2.3693	21.6522	5.5544	9.1681	8.1999	12.0838
2	7.9131	9.3813	7.6553	8.9725	9.1276	8.5882
3	3.3406	9.5463	5.4354	9.0099	8.7704	4.7512
4	11.5341	8.1548	5.5472	9.3074	8.0422	3.9884
5	4.2335	15.0218	5.7794	9.1677	7.8680	6.2654
6	3.2988	15.6121	5.7700	9.1559	7.8192	11.5714
7	9.4303	8.4669	5.8065	9.1180	9.6999	4.2183
8	2.2772	9.8217	6.1050	9.0190	9.0931	5.5352
9	6.4085	28.7995	5.7913	9.1148	8.2983	6.8333
10	4.0160	8.0488	5.7208	9.5146	7.9660	5.7379
Avg.	**5.4822**	**13.4505**	**5.9165**	**9.1548**	**8.4885**	**6.9573**
Std.	**3.1973**	**6.9683**	**0.6380**	**0.1594**	**0.6451**	**2.8947**

b. Vales are on the scale of $1e^{-5}$

Based on the training and testing model that were described in the previous section, we have predicted the relative tendency growth between DJIA and its five randomly picked component stocks in the fifth trading day by providing their price indexes of the past four continuous trading days. Our final prediction is a positive one if DJIA relatively performs better than stock X in the next day, and negative otherwise. Furthermore, the accuracy of our prediction is calculated by comparing the difference between our final prediction and the actual tendency.

The MSE (Mean Squared normalized Error) and the corresponding standard deviation of all six network training models are presented in Table III. Table IV illustrates the MSE and corresponding standard deviation of the networks according to the testing data sets. The two tables show that our simulation network models perform very well in prediction, which is demonstrated in Table V.

TABLE V. ACCURACY OF OUR PREDICTION IN TERMS OF PERCENTAGE

Run	DJI	ATT	CAT	INTC	JPM	MMM
1	100.00	94.58	95.07	89.65	85.22	98.02
2	100.00	95.07	95.07	**92.61**	86.20	98.02
3	100.00	96.05	96.05	**92.61**	84.72	97.53
4	100.00	**95.56**	95.56	89.16	84.72	98.52
5	100.00	95.07	96.05	92.11	**87.68**	**99.01**
6	100.00	95.07	94.58	91.62	84.72	**99.01**
7	100.00	95.07	93.10	90.64	86.69	96.55
8	100.00	95.07	94.58	90.64	84.72	98.52
9	100.00	**95.56**	95.56	90.14	82.75	97.53
10	100.00	94.58	**96.55**	90.64	84.72	98.52
Avg.	**100.00**	**95.17**	**95.22**	**90.98**	**85.22**	**98.13**
Std.	**0.0000**	**0.4526**	**0.9865**	**1.2077**	**1.3540**	**0.7631**

c. DJI is the base referencing stock

Table V illustrates the accuracies of our prediction on relative tendency growth among stocks. We can see that the best performance of our system is on predicting the relative tendency growth between DJIA and MMM, which is 99.01%. For the other 4 component stocks, two of them have more than 95% accuracy, one has 92.61%, and even the worst performance has an accuracy of 85.22% on average over the ten runs. With such predictions, we can easily obtain the information of how well the component stocks will perform relative to the DJIA, or any other stocks.

VI. CONCLUSION AND FUTURE WORK

In this article, we presented a novel prediction system for stock trading. The system utilizes artificial neural networks to forecast the tendency of different stocks and their relative tendency growth. We have selected Dow Jones Industrial Average as the reference index and compared its relative tendency growth with five of its component stocks during the period from 02-Feb-2004 to 26-Feb-2008 (1024 trading days). According to EMH, stock prices should follow a random walk pattern and therefore cannot be predictable much past 50%. However, by using our system, we achieve 99.01% accuracy in predicting the relative tendency growth among different stocks. In the future, we expect to utilize different machine learning algorithms to test the performance of our system, in terms of prediction accuracy. Furthermore, we also plan to apply our system to other famous stocks and indices, such as the NASDAQ Composite Index, the NYSE Composite Index, and the S&P 500 Composite index.

REFERENCES

[1] Fama, E. F., 1965. The Behaviour of Stock Market Prices. Journal of Business, 38, 34-105.

[2] Fama, E. F., Fisher, L., Jensen, M. C., & Roll, R. W., 1969. The Adjustment of Stock Price to New Information. International Economic Review, 10(1), 1-21.

[3] Fama, E. F., 1970. Efficient capital markets: A review of theory and empirical work. Journal of Finance, 25, 383-417.

[4] Fama, E. F., 1991. Efficient Capital Markets: II. Journal of Finance, 46(5), 1575-1617.

[5] Louis Bachelier – http://en.wikipedia.org/wiki/Louis_Bachelier

[6] Cootner, P. H. (1964). The Random Character of Stock Market Prices. MA: MIT Press.

[7] Alexander, S. S. (1961). Price Movements in Speculative Markets: Trends or Random Walks. Industrial Management Review, May, 7-26.

[8] Jensen, M. C. (1978). Some Anomalous Evidence Regarding Market Efficiency. Journal of Financial Economics, Vol. 6, 95-102.

[9] Gallagher, L. A., & Taylor, M. P., 2002. Permanent and Temporary Components of Stock Prices: Evidence from Assessing Macroeconomic Stocks. Southern Economic Journal, 69, 245-262.

[10] Lo, A. W. & MacKinlay, A.C., 1997. Stock Market Prices Do Not Follow Random Walks. Market Efficiency: Stock Market Behaviour in Theory and Practice, 1, 363-389.

[11] Kavussanos, M. G. & Dockery, E., 2001. A multivariate test for stock market efficiency: the case of ASE. Applied Financial Economics, vol. 11, no. 5, 573-579, 1 October.

[12] Butler, K. C., & Malaikah, S.J., 1992. Efficiency and inefficiency in thinly traded stock markets: Kuwait and Saudi Arabia. Journal of Banking & Finance, Volume 16, Issue 1, 197-210, February.

[13] Kirkpatrick and Dahlquist Technical Analysis: The Complete Resource for Financial Market Technicians (Financial Times Press, 2006), page 3.

[14] Graham, B., & Dodd, D., 1934. Security Analysis. New York and London: McGraw-Hill.

[15] B. Qian and K. Rasheed, Hurst Exponent and Financial Market Predictability". IASTED Conference on Financial Engineering and Applications – 2004

[16] W. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, 7, 1943,:115 - 133.

[17] F. Rosenblatt, The Perceptron: a probabilistic model for Information storage and organization in the brain, Psychological Review, 65(6), 1958, 386-408.

[18] F. Rosenblatt, Principles of neurodynamics, (Washington D.C.: Spartan Press, 1961). [18] M. Minsky & S. Papert, Perceptrons (Cambridge, MA: MIT Press, 1969)

[19] D.E. Rumelhart, G.E. Hinton & R.J. Williams, Learning internal representations by error propagation, in Parallel distributed processing, 1, (Cambridge, MA: MIT Press, 1986)

[20] J. Yao, C.L. Tan & H. Poh, Neural networks for technical analysis: a study on LKCI, International journal of theoretical and applied finance, 2(3), 1999, 221-241

[21] Peters, E. E., 1994. Fractal market analysis: applying chaos theory to investment and economics. New York: Wiley.

[22] Walczak, S., 2001. An empirical analysis of data requirements for financial forecasting with neural networks. Journal of management information systems, 17(4), 203-222.

[23] Takens, F., 1980. Detecting strange attractors in turbulence. In Rand, A., & Young, L. S. (Ed.), Dynamical system and turbulence, lecture notes in mathematics, 898(Warwick) 1980., Berlin: Springer.