Modeling the Behavior of the S&P 500 Index: A Neural Network Approach

Mary E. Malliaris
Department of Management Science
Loyola University Chicago

Abstract

The October 1987 stock market crash challenged the prevailing financial models of a random walk and led to the emergence of a new and competing model of stock price time series. This new approach supports a non-random underlying structure and is labeled chaotic dynamics. If a neural network can be constructed which determines market prices better than the random walk model, it would support those who claim that they have found statistical evidence that a chaotic dynamics structure underlies the market. This paper constructs a neural network which lends support to the deterministic paradigm.

1. Introduction

For a long time, it was believed that price changes in the stock market were, to a large extent, random [4]. However, the October 1987 stock market crash challenged the then prevailing financial models of a random walk. A new and competing model of stock price time series has emerged which supports a non-random underlying structure in the market. This approach is labeled chaotic dynamics.

The chaotic dynamics approach is a deterministic model which yields a time series behavior that appears random when in fact such a series is generated by a nonlinear deterministic equation. Sometimes small input changes can produce very divergent outputs, causing a seemingly orderly system to become chaotic. The chaos appears to be random and unpredictable while it is actually following strict mathematical principles. Although deterministic, chaotic dynamics, when graphed, looks like a random walk. Preliminary statistical evidence has not succeeded in rejecting the presence of chaos in the S&P 500 Index series. The possibility that the underlying dynamics of the S&P 500 might follow a model of low order, nonlinear, deterministic chaos, motivates the search for a neural network which can indicate the existence of such a structure.

Neural networks, built to pattern the way a brain learns, do not require that the relationships among variables be specified in advance. If a neural network can be constructed which determines market prices, this would imply that the network has discovered an underlying structure in the data. Such a result would challenge the random walk hypothesis and would support those who claim that they have found statistical evidence that a chaotic dynamics structure underlies the market. [6]

The implications of this investigation are quite significant for several reasons. First, the neural network results will give evidence as to the appropriateness of one of the two alternative paradigms, random walk or chaotic. Second, support for the chaotic paradigm would imply that active management of an S&P 500 portfolio is possible since the S&P follows a non-linear deterministic model. And third, if a neural network can outperform the random walk, then researchers would be encouraged to search for expressions linking the unknown but deterministic pattern of the S&P 500 to the explanatory variables.

2. Review of the three models

In its simplest formulation we define the sequence of prices, denoted by $\{p(t): t = 0, 1, 2, ...\}$, to follow a random walk if $p(t+1) = p(t) + \varepsilon(t+1)$, where $\varepsilon(t+1)$ is the value obtained from sampling with replacement from a certain distribution with a given population mean μ and a variance of σ^2. This equation expresses tomorrow's price as a random departure from today's price, or equivalently, the price change between today and tomorrow, i.e. $p(t+1) - p(t)$, as random. It is usually assumed that $\mu = 0$. The random walk model is utilized to convey the notion that stock prices cannot be systematically forecasted [9].

The efficient market hypotheses was developed to rationalize the random walk behavior claiming that the current price **p(t)** fully and correctly reflects all relevant information and because the flow of information between now and next period cannot be anticipated, price changes are serially uncorrelated. Though numerous studies have confirmed market efficiency, other studies have rejected it. The rejections of the random walk paradigm were considered to be anomalies by efficient hypothesis researchers. However, the October 1987 stock market crash caused a serious reevaluation of the efficient market hypotheses. Shleifer and Summers [12] are quite critical and they claim that "the stock in the efficient market hypothesis -- at least as it has been traditionally formulated -- crashed along with the rest of the market on October 19, 1987".

If the random walk model is not a satisfactory description of stock price behavior and if prices move without any obvious change in the fundamentals of the economy, what methodological alternatives exist to explain the observed price patterns? To answer this question, a handful of quantitative economists investigated the deterministic methods of Ruelle and Takens [10] who studied the physical problem of turbulence. These authors and the numerous physicists who followed them developed a very active field of current research called chaotic dynamics. Chaotic dynamics yields a time series behavior that appears random when in fact such a series is generated by a nonlinear deterministic equation of high degree. The chaos in the system appears to be random and unpredictable while it is actually following strict mathematical principles.

Consider a real-valued function $\mathbf{f{:}R \rightarrow R}$. We are interested in the time series generated by this function starting from some arbitrary $\mathbf{x_0} \in \mathbf{R}$. Denote by $\mathbf{f^2} \equiv f[f(x)] \equiv f \circ f(x)$ where o means composition and in general let $\mathbf{f^n = f \circ f \circ \ldots \circ f(x)}$ mean **n** compositions. The time series takes the values $x_o, f(x_o), f^2(x_o), \ldots, f^n(x_o), \ldots$, for $\mathbf{t = 0, 1, 2, \ldots, n}$. For this to describe a chaotic function it must satisfy three requirements.

First it must sample infinitely many values. The second requirement is sensitive dependence on initial conditions. This condition says that there are time series that start very close to each other but diverge exponentially fast from each other. The third requirement is that the periodic points of **f** are dense in **R**.

The methodology for detecting chaotic dynamics in stock price changes frequently uses the Grassberger and Procaccia [3] correlation integral to compute the correlation dimension. Several studies have computed the correlation dimension for the S&P 500 Index. For example, Scheinkman and LeBaron [11] concluded that the correlation dimension for the S&P 500 Index appeared to be about 6, implying that such an index has nonlinear structure. The investigation of chaotic systems can become extremely complicated. But the evidence of some unknown underlying structure motivates the search for nonlinear behavior through neural networks [14].

A neural network uses an abundance of input data that require categorizing and interpreting. It is not necessary to specify an underlying structure, since the network infers the patterns by generalizing from the interaction of the inputs. Neural networks are structured layers of nodes, weights and connections. The layers include an input layer, an output layer, and one or more interior layers called the middle or hidden layers. Signals travel through the network from the input layer to the hidden layers to the output layer. Each node of each layer is connected to each node of the next layer. Each connection between neurons has a numerical weight, either positive or negative, associated with it which is multiplied times the data coming from the previous layer node to which it is connected. This weight reinforces or inhibits the effect of the previous node on the next layer node. The node applies a transfer function to the sum of the weighted inputs and computes one output signal.

Learning in a backpropagation neural network involves two repeated phases: the forward phase and the backward phase. During the forward phase, the input is sent forward through the network, generating an output value for the final layer node. The difference between what the actual output should be and the network's output is computed. In the backward phase, the computed error is used and weights are changed in proportion to the error times the input node signal. After weight adjustment, the data is again sent through the network, and the process continues until the difference of each output of the network and the actual value is below some specified level. For a detailed explanation of the training process, see Malliaris and Salchenberger [8].

3. Data

Weekly data have been collected from each Friday for two years, 1989 and 1990, on ten variables, including: the S&P 500 closing Index, the three month Treasury Bill interest rate, the thirty year Treasury Bond interest rate, weekly New York Stock Exchange volume, Money Supply as measured by both M1 and M2, Price/Earnings ratio, Gold price, Crude Oil price, and the CBOE put/call ratio. The stock market is influenced by expectations of the traders, fundamental measures of economic activity and technical factors such as trading volume. Some representative references that discuss the selection of these variables are [1], [2], and [7].

A generally recommended method for assessing the true unbiased amount of error in a neural network model is the use of cross-validation [13], [5]. To use cross-validation, the data set must be divided into k distinct sets of about the same size. Each set is used independently for testing while the remaining data is utilized for training the network. Each training and testing of the network will result in a final error amount in each set. The average of these errors over all the k sets is an almost unbiased estimator of the true error rate [13].

4. Methodology

The efficient market hypothesis claims that the best estimate of a value for a following period is the same as the value in the current period. To have a baseline amount against which to compare the performance of the neural network, we calculate the difference between the next week's actual and expected values and use this value as the number to "beat" if the random walk hypothesis is to be rejected. The results are calculated for the same ten testing periods used for the network cross-validation and will be shown as the Mean Absolute Deviation (MAD), Mean Square Error (MSE) and as a correlation between expected and actual output. After using the constructed neural network to generate output for the same weeks, the same statistics are also calculated for the network and actual differences. For the neural network to do a better job predicting than the random walk hypothesis, these comparison statistics must outperform those generated previously more than 50% of the time.

The neural network was built and refined using California Scientific Software's Brainmaker v2.53 and Genetic Training Option, run on an Intel 486/50. When setting up a neural network, one must decide how many nodes to have in the input and hidden layers. Because a neural network cannot "remember" data from a previous row in the sense of a time series, it is important to include in each row enough lags to give the network this knowledge. Each of the data input variables was lagged twice, tripling the number of input nodes. Input nodes were also included for the week of the month and the month of the year. The output node was the following Friday's value of the S&P Closing Index. That is, the network was structured to give an estimate for the following Friday's S&P 500 Index.

In order to determine the optimal number of nodes in the hidden layer or layers, Brainmaker's Genetic Training Option was implemented. This program tests all possible networks within user-specified parameters. The number of hidden layers was varied between 1 and 2, with the number of nodes in each layer varying from 2 to 45. The training tolerance, originally set at the default value of .1, is a number Brainmaker uses to compare network output to actual values (data has been scaled to be between 0 and 1). If the difference between the two is less than the tolerance, the fact is classified as "good". When the network can judge all the training facts as good, it stops training, i.e., it has converged. Lowering the tolerance forces the network to work a little harder: Convergence will take longer but the final error should be smaller and the correlation between output and actual values in the testing set may be higher. The training tolerance was lowered to .07. That is, no network output would be classified as good if its scaled absolute difference from the scaled actual output was greater than .07.

The Genetic Training Option records statistics on each network configuration which allow for comparison between the configurations. The network with the lowest Root Mean Square Error and the highest correlation had the configuration of 2 hidden layers with 24 nodes in the first hidden layer and 8 nodes in the second hidden layer.

Once the number of layers and nodes per layer was decided, the weights assigned to each node were adjusted to give the best performance. Initially, weights are random; they are refined through an evolutionary process. This was done using Brainmaker's Genetic Training Option to genetically evolve the trained network. In this process, the weights are mutated and crossed over, based on theories of genetic evolution, to see whether a child network can be evolved with the same node structure as the parent, but with superior weights. The mutation rate was set at 10%, the crossover rate at 50% and each child network was run 100 times to polish the weights. The resulting best weights were saved and used as the weights of the network.

The data set was randomized and divided into 10 sets of data, each with 10 rows. This was done in order to use cross-validation to better estimate the true error. Each set was used, in turn, as a testing set, while the network was trained on the remaining data. Using cross-validation techniques, the RMS errors and the correlations between network output and actual S&P Index values were calculated for each of the ten testing sets. The average RMS error, the almost unbiased estimate of the true error, across all the sets was 0.046 (Brainmaker's result on scaled data), with an average correlation of 0.964.

5. Analysis

The MAD, MSE and correlation for each of the ten non-overlapping sets were calculated, comparing the actual value of the S&P to the efficient market hypothesis, which expects the next period's value to be equal to today, and to the output from the neural network. The neural network output for each of the sets was generated on the testing set using the network which had been trained on the corresponding training set. The results are shown in Table 1. As can be seen for each testing set, the neural network outperformed the random walk model in each category. The average MAD for the network was 3.167, as opposed to 6.188 for the random walk. The average MSE for the network was 15.609, as opposed to 41.32 for the random walk.

The strong evidence that in ten out of ten sets of data covering a span of two years, neural networks have outperformed the random walk model can be interpreted to be supportive of the deterministic structure of the stock market returns during our sample period. Results of this nature are encouraging to researchers who wish to develop deterministic theories which eventually may replace the existing probabilistic paradigm.

We have argued, as have other financial researchers, that the October 1987 stock market crash, to this date, has not been reconciled with the efficient market hypotheses. While numerous researchers have criticized the random walk behavior and have sought evidence of a deterministic structure in stock market returns, no theoretical framework currently exists that describes the stock returns in a deterministic way. If such a structure were to be identified, phenomenal arbitrage opportunities would become available. Although the development of such deterministic models may take a long time to be developed, results such as these shown here present a challenge to efficient market theorists.

References

1. Estrella, A. and G. Hardouvelis, (1991), "The Term Structure as a Predictor of Real Economic Activity", The Journal of Finance, XLVI, 2, pp. 555-576.

2. Fama, E., (1990), "Stock Returns, Expected Returns, and Real Activity", The Journal of Finance, XLV, 4, pp. 1089-1108.

3. Grassberger, P. and I. Procaccia, (1983), "Measuring the Strangeness of Strange Attractors", Physics, Vol. 9-D, pp. 189-208.

4. Kendall, M. (1953), "The Analysis of Economic Time Series, I", Journal of the Royal Statistical Society (Series A), CXVI, pp. 11-25.

5. Kuhn, G. and N. Herzberg, (1991), "Some Variations on Training of Recurrent Networks", in Neural Networks, Theory and Applications, Academic Press, Inc., Boston.

6. Malliaris, M. (1993), "The S&P 500 Index: A Neural Network Supports the Deterministic Paradigm", working paper.

7. Malliaris, A. and M. Malliaris, (1992), "Interest Rates and Inflation, A Continuous Time Stochastic Approach", Economics Letters, Vol. 37, pp. 351-356.

8. Malliaris M. and L. Salchenberger, (1993), "Beating the Best: A Neural Network Challenges the Black-Scholes Model", Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications, March 1-5, 1993, pp. 445-449.

9. Roberts, V., (1959), "Stock-Market 'Patterns' and Financial Analysis: Methodological Suggestions", The Journal of Finance, Vol. 14, No. 1, pp. 1-10.

10. Ruelle, D., and F. Takens, (1971), "On the Nature of Turbulence", Communications of Mathematical Physics, Vol. 20, pp. 167-192.

11. Scheinkman, J., and B. LeBaron, (1989), "Nonlinear Dynamics and Stock Returns", Journal of Business, Vol. 62, pp. 311-337.

12. Sheifer, A., and L. Summers, (1990), "The Noise Trader Approach to Finance", The Journal of Economic Perspectives, 4, pp. 19-33.

13. Weiss, S., and C. Kulikowski, (1991), Computer Systems That Learn, Morgan Kaufmann Publishers, Inc., San Mateo, Ca.

14. White, H.,, (1988), "Economic Prediction Using Neural Networks: The case of IBM Stock Prices", Proceedings of the IEEE Second International Conference on Neural Networks, II, pp. 451-458.

15. White, H., (1989), "Neural-network Learning and Statistics", AI Expert, 4,12,p. 52.