Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 171–180
Expert Systems with Applications
www.elsevier.com/locate/eswa

A fusion model of HMM, ANN and GA for stock market forecasting

Md. Rafiul Hassan *, Baikunth Nath, Michael Kirley

Computer Science and Software Engineering, The University of Melbourne, Carlton 3010, Australia

Abstract

In this paper we propose and implement a fusion model by combining the Hidden Markov Model (HMM), Artificial Neural Networks (ANN) and Genetic Algorithms (GA) to forecast financial market behaviour. The developed tool can be used for in depth analysis of the stock market. Using ANN, the daily stock prices are transformed to independent sets of values that become input to HMM. We draw on GA to optimize the initial parameters of HMM. The trained HMM is used to identify and locate similar patterns in the historical data. The price differences between the matched days and the respective next day are calculated. Finally, a weighted average of the price differences of similar patterns is obtained to prepare a forecast for the required next day. Forecasts are obtained for a number of securities in the IT sector and are compared with a conventional forecast method.

Keywords: Forecasting; Stock market; Hidden Markov Model; Artificial Neural Network; Genetic Algorithm

1. Introduction

Availability of assets and money can assure us a secured and comfortable life (Lawrence, 1997). It is no surprise that so much attention has been devoted to the analysis and prediction of future values and trends of the financial market. A variety of forecasting methods have been proposed and implemented. Each method has its own merits and limitations.

In recent years, with the introduction of online trading, the stock market has become one of the avenues where even small investors can earn good profits. It would therefore be quite appealing if we can predict the market behaviour accurately, so that investors can decide when and where to invest their money. However, because of the high volatility of the underlying laws behind the financial time series, it is not an easy task to build such a forecasting model. There are many complex events that can affect the stock market, e.g. business cycles, monitory policies, interest rates, political situations, etc. A number of forecasting methods are available, but most of the models have their own merits and limitations. Specially the conventional statistical techniques are constrained with the underlying seasonality, non-stationarity and other factors (Tambi, 2005). Furthermore, use of conventional statistical methods without technical expertise becomes difficult. In this study, we propose a forecasting tool, which is easy to use and interpret.

A Hidden Markov Model (HMM) is a widely tool to analyse and predict time series phenomena. HMM has been used successfully to analyse various types of time series including DNA sequence analysis (Cheung, 2004), Speech Signal recognition (Xie, Andreae, Zhang, & Warren, 2004), ECG analysis (Coast, Stern, Cano, & Briller, 1990) etc. In an earlier study (Hassan & Nath, 2005) HMM has been used to generate one-day forecasts of stock prices in a novel way.

In this paper we propose and develop a fusion model combining the HMM used in Hassan and Nath (2005) with an Artificial Neural Network (ANN) and a Genetic Algorithm (GA) to achieve better forecasts. In our model, ANN is used to transform the input observation sequences of HMM and the GA is used to optimize the initial parameters of the HMM. This optimized HMM is

* Corresponding author. Present address: Department of Computer Science and Software Engineering, The University of Melbourne, 111, Barry Street, Victoria 3010, Australia. Tel.: +61 3 83441426; fax: +61 3 93481184.

E-mail addresses: mrhassan@cs.mu.oz.au, hassan.rafiul@gmail.com (Md.R. Hassan).

0957-4174/$ - see front matter
doi:10.1016/j.eswa.2006.04.007

then used to identify similar data pattern from the historical dataset. Two alternative techniques are examined for constructing the one day forecast value based on the HMM identified patterns. In the first approach, we interpolate the neighboring values of the data patterns to obtain the forecast value as reported in Hassan and Nath (2005). In the 2nd approach of the proposed model, we do not interpolate the neighbouring values as is done in Hassan and Nath (2005). As a substitute we find a number of similar data patterns from the historical data and then use a weighted average of the difference values between neighbouring data items. This value is then added to the current value of the variable of interest to obtain a one time unit forecast value. Details of the proposed fusion method are provided in Section 3.

The remainder of the paper is organized as follows: Section 2 describes the background, Section 4 presents the experimental results and analysis and finally Section 5 reports discussion followed by a brief conclusion in Section 6.

2. Background

Over the last few decades, a large number of studies have proposed and developed different methods to analyse and forecast stock market activity. In what follows a brief discussion of the key studies are provided.

ANN represents one widely used soft computing technique for stock market forecasating. Apparently, (White, 1988) first used Neural Networks for market forecasting. He used the IBM daily common stock returns and found that the training results were over optimistic. Kimoto, Asakawa, Yoda, and Takeoka (1990) have reported on the effectiveness of alternative learning algorithms and prediction methods using ANN, when for developing a Tokoyo Stock exchange prices index prediction system. In other work, Chiang, Urban, and Baldridge (1996) have used ANN to forecast the end-of-year net asset value of mutual funds. Trafalis (1999) used feed-forward ANN to forecast the change in the S&P(500) index. In that model, the input values were the univariate data consisting of weekly changes in 14 indicators. Choi, Lee, and Rhee (1995) forecasted the daily direction of change in the S&P(500) index futures using ANN. Despite the wide spread use of ANN in this domain, there are significant problems to be addressed. ANNs are data-driven model (White, 1989; Ripley, 1993; Cheng & Titterington, 1994), and consequently, the underlying rules in the data are not always apparent (Zhang, Patuwo, & Hu, 1998). Also, the buried noise and complex dimensionality of the stock market data makes it difficult to learn or re-estimate the ANN parameters (Kim & Han, 2000). It is also difficult to come up with an ANN architecture that can be used for all domains. In addition, ANN occasionally suffers from the overfitting problem (Romahi & Shen, 2000).

Fuzzy logic, based on expert knowledge, is an example of another tool used in the stock market forecasting domain. However, where (when) expert knowledge is not available, there are alternative methods to extract rules from the dataset. Fuzzy logic may be used for stock market forecastaing either independently or hybridized with other methods. For instance, Romahi and Shen (2000) have developed an evolving rule based expert system, where fuzzy logic and the rule induction were merged together to obtain a promising method to forecast financial market activity. Another interesting model is EFuNN (Abraham, Nath, & Mohanthi, 2001). This model is a hybridization of Fuzzy logic, ANN and the Evolutionary computation, which has been used effectively to forecast and analyse the trend of Nasdaq-100 index values and six other companies listed in the Nasdaq-100 index. Though applications of Fuzzy logic for stock market forecasting have been found, still the limitation reported with tool remains. This is the requirement of expert knowledge prior to designing fuzzy methods for forecasting, which requires further research.

A number of statistical tools are available to make forecasts, based on past time series data. The traditional statistical approaches are: Box-Jenkins or ARMA method (Box & Jenkins, 1976; Pankratz, 1983), the threshold Autoregressive model (Tong & Lim, 1980), the Autoregressive Conditional Heteroscedastic (ARCH) (Engle, 1982) and Multiple Linear Regression model.

Henry (1993), used the ARIMA model to forecast the daily close and morning open price in Hong Kong stock market. In other work Raymond (1997), applied ARIMA model to forecast the real state market. However, with ARIMA models, problems arise when the variance in the time series increases or when nonlinear processes exist in the time series (Kolarik & Rudorfer, 1994).

2.1. Stock market forecasting using HMM

A Hidden Markov Model, which is built on the probabilistic framework for modelling a time series of multivariate observations, is used by Hassan and Nath (2005) to forecast stock prices due to its strong statistical foundation, ability to handle new data robustly, the computational efficiency to develop and the ability to predict similar patterns efficiently. There are three parameters: transition matrix (A), observation emission matrix (B) and the prior probability matrix (π) of a HMM, which represent the overall HMM model λ. In that study, the popular Baum–Welch algorithm was employed to re-estimate these parameters so that the model best fits the training dataset. In doing so, a method called forward-backward (Rabinar, 1989) algorithm is adopted to compute the probability $P(O|\lambda)$ (Logarithm of this probability is called log-likelihood value or likelihood value) of observation sequence $O = O_1, O_2, \ldots, O_T$, for the given model λ.

In Hassan and Nath (2005) an initial HMM structure was built consisting of four states (with four input features: opening, high, low and close prices). Initially the parameter values were chosen randomly. The HMM was then trained using the training dataset so that the values of A, B and π

are re-estimated to be best suited with the training dataset. The next step of this method was to locate the past day(s) stock behaviour which would be similar to that of the current day 'c'. To do this, the authors obtained the likelihood value for the observation sequence on day 'c'. Each of the observation sequence was built using the daily open, high, low and close price (in that study these four variables were used as predictors for the next day's stock price). For better clarification, let us assume the likelihood value for observation sequence on day 'c' is 'L_c'. Now, from the historical dataset using the HMM those observation sequences are located which would produce the same 'L_c' or close to the 'L_c' value. Say, the HMM found the mth day's observation sequence which would produce the same likelihood value 'L_c'. Now, the price difference, '*diff*', of the variable of interest, on day 'm' and '$m+1$' is calculated. Using Eq. (1) the forecast value of the variable of interest on day '$c+1$' (if day 'c' represents the current day '$c+1$' would be next to current day) is obtained:

$$\text{Forecast value (of the interested variable) on day `}c+1\text{'} \\ = \text{The value (of the interested variable) on day `}c\text{'} + \text{`}\mathit{diff}\text{'} \quad (1)$$

Hassan and Nath (2005) used multiple variables, instead of only one variable, to facilitate the forecasting accuracy. As some literature suggest that, by combining related time series helps to predict the time series with greater accuracy (Yang & Liu, 2001). In Yang and Liu (2001) model, an ANN was used to predict the multivariate time series combined from open, high, low and close SSE (Shanghai Stock Exchange) index series. However, the HMM used in the study (Hassan & Nath, 2005) may not efficiently recognize the similar observation sequence from the past dataset, as HMM suffers from poor parameter initialization. The HMM may not be well trained if the training observation sequence does not suit properly with the parameters chosen initially to build the HMM structure. Another issue of the approach (Hassan & Nath, 2005) which may be modified to improve the forecast accuracy is that, an interpolation is done by locating only a single observation sequence. (Note: we will refer to the model presented in Hassan & Nath (2005) as HiMMI in the subsequent sections). To overcome the above mentioned limitations of the HiMMI, we develop a fusion model combining ANN and GA with HiMMI.

3. Overview of the fusion model

To forecast stock prices, a fusion model of HiMMI, GA and ANN is developed here in this study. As noted previously, the performance of the HMM in HiMMI is dependent on how well its parameters are re-estimated for the given observation sequences. In the proposed fusion model, an ANN is employed as a black-box to introduce noise to the observation sequences so that they may be better fitted with the HMM. The GA then is used to find out the optimal initial parameters for the HMM given the transformed observation sequences. This fusion model is then likely to find a number of alternative data items from the historical data, which exhibit similar stock market behaviour to that of the current day. Then a weighted average of the price differences for the identified data items is calculated. This weighted average is added to the current day price. The obtained value is the one day forecast value. Fig. 1 shows the block diagram of the model. There are two phases in this model:

1. Optimisation of HMM in HiMMI.
2. Obtaining the forecast using weighted average.

These phases are described in detail below.

3.1. Optimisation of HMM

In the fusion model, the ANN and GA processes are executed iteratively until a specified GA stopping criterion is reached. The following subsections describe the internal links of the fusion model in more detail.

3.1.1. Linking of the ANN with the HMM

There are strong dependencies between the observation sequences and the re-estimation criteria of the HMM. The optimization (re-estimation) process of the HMM parameters may be more efficient, if we could translate the observation sequences which would be better suited

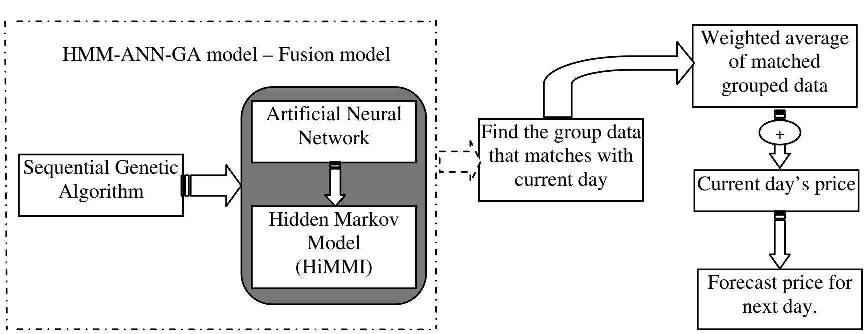

Fig. 1. Block diagram of the fusion model.

for the HMM. For instance, to obtain an optimized HMM for pattern matching, Bengio, Mori, and Kompe (1992) used ANN to transform the actual observations and then the transformed observations were fed into the HMM as an input vector. After optimizing the HMM for the transformed observations, significant improvement was achieved. Following the study (Bengio et al., 1992) we explain in more detail, how the observation sequences and the optimization of HMM are related to each other.

Let us assume that the transformed observation vector at a point in time 't' is Y_t, which is fed into the HMM as an input observation (input sequence). Thus the properties of the HMM are as follows:

Y_1^T	the whole observation sequence
Y_t	an input observation at time 't'
a_{ij}	the transition probability from state 'i' to state 'j'
b_i	the emission probability from state 'i'

and the probability that the HMM generates Y_t in state S_t at time t is denoted as

$$b_{i,t} = P(Y_t|S_t = i)$$

with the alluded properties, the algorithms described in Rabinar (1989) is used in a recursive way to compute the following probabilities for the partial observation sequences up to time t with some boundary conditions assumed:

$$\alpha_{i,t} = P(Y_1^t \text{ and } S_t = i|\text{HMM}) = b_{i,t}\sum_j a_{j,i}\alpha_{j,t-1} \tag{2}$$

$$\beta_{i,t} = P(Y_{t+1}^T|S_t = i \text{ and HMM}) = \sum_j a_{i,j}b_{j,t+1}\beta_{j,t+1} \tag{3}$$

The re-estimation or the optimization of the HMM parameters can be done considering different criteria and uses the probabilities obtained in Eqs. (2) and (3). Among them, the maximum likelihood (ML) and maximum mutual information (MMI) are the most popular. Modelling with these two criteria is discussed in Nadas, Nahamoo, and Picheny (1988). For the case of MLE (Maximum Likelihood Estimation) for a HMM where the $b_{i,t}$ is assumed as Gaussian mixtures as follows:

$$b_{i,t} = \sum_k \frac{Z_k}{\left((2\pi)^n|\sum_k|\right)^{1/2}} \times \exp\left(-\frac{1}{2}(Y_t-\mu_k)\sum_k^{-1}(Y_t-\mu_k)^T\right) \tag{4}$$

where, n is the number of observation features of the HMM. The transition probabilities a_{ij}, normal distribution mean vectors μ_k, covariance matrices $\sum_k$, and gains Z_k can be estimated as in Rabinar (1989). As the optimization criterion 'C' used for the HMM, depends on Y (the output produced by the ANN) it is possible to express 'C' as a function of Y and derive the following equation, using the chain rule (Bengio et al., 1992):

$$\frac{\partial C}{\partial Y_{jt}} = \sum_i \frac{\partial C}{\partial b_{i,t}}\frac{\partial b_{i,t}}{\partial Y_{jt}} \tag{5}$$

Eq. (5) shows that the optimization criterion 'C' depends on the mutual corresponding (or we can say the degree of suitability between the input sequences and the observation emission probabilities) between the input sequence Y_{jt} (jth observation at time 't') and the observation emission probabilities $b_{i,t}$.

In this study, we consider the dependency of the optimization criterion of HMM on the observation sequence but do not follow the global optimization methodology described in Bengio et al. (1992). As an alternative, we employ a GA tool to obtain optimized initial parameter values of the HMM, which after the training, best fits with the transformed observation sequences Y. This process is executed until a possible best combination of ANN and optimized HMM is found.

So, in order to combine ANN and HMM systems we follow these steps while Figs. 2 and 3 show the integration process:

1. Create an ANN structure randomly consisting of 'n' number of input nodes and 'n' number of output nodes (where 'n' is the number of predictors).
2. Initialize the weights of the ANN randomly.
3. The actual observation vectors are fed into the ANN as input.
4. The output vector Y_t produced by the ANN at time 't' is fed to the HMM as input observation vector.

Fig. 4 shows the actual observation sequences and the transformed observation sequences using an ANN. The next subsection describes how GA is used to optimize the combination of ANN and HMM.

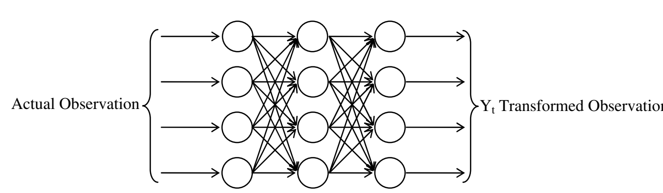

Fig. 2. A feed-forward neural network to transform the inputs.

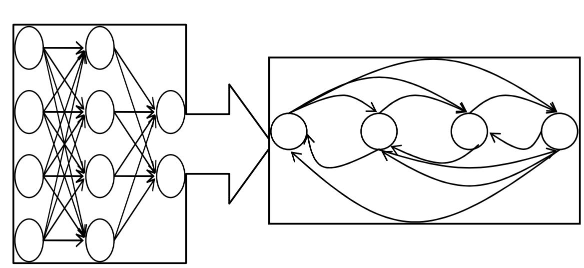

Fig. 3. Integrated ANN-HMM model.

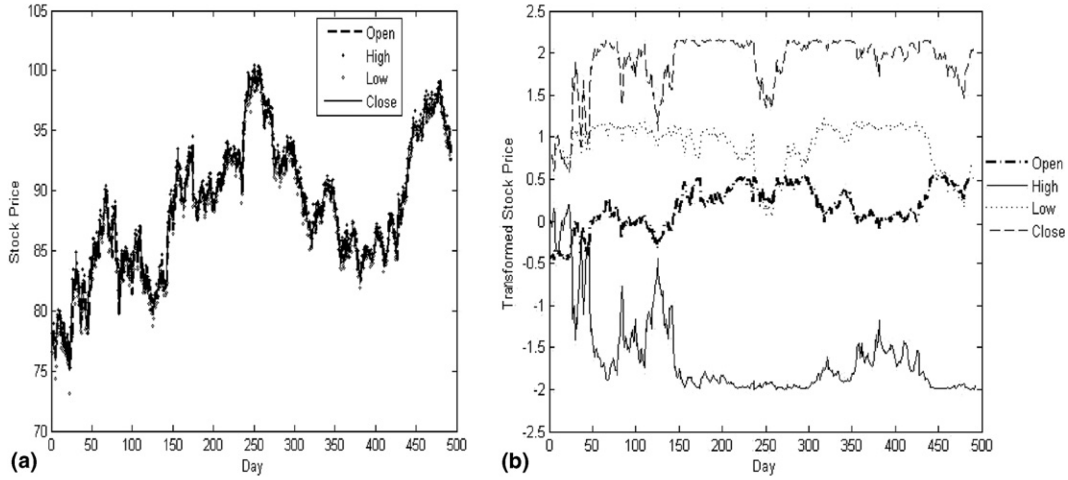

Fig. 4. (a) The actual observation sequences and (b) the transformed observation sequences.

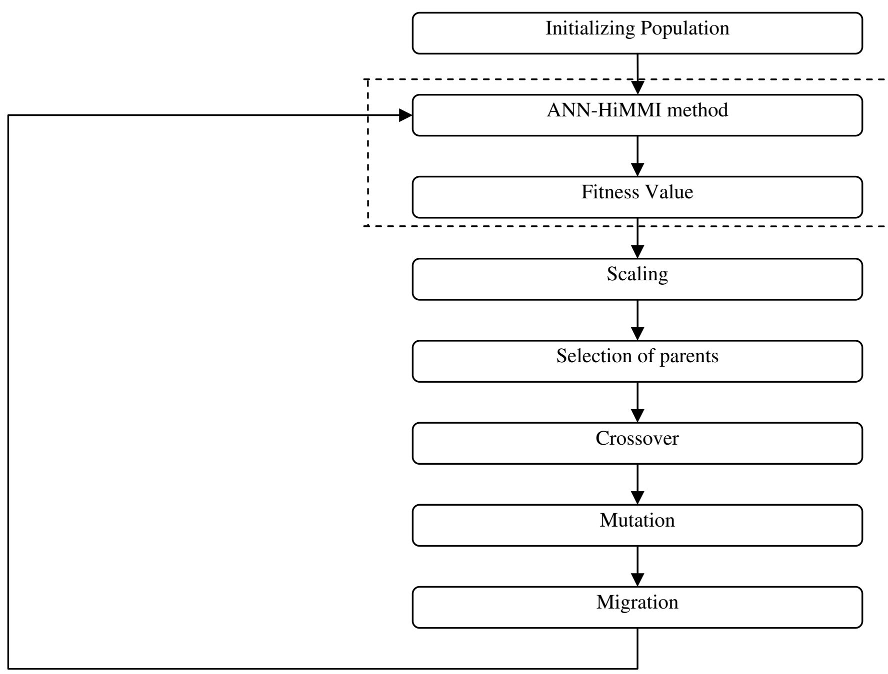

Fig. 5. A flow-chart showing how GA works for optimization of the parameters of ANN-HiMMI method.

3.1.2. Linking of the GA with HMM and optimisation of initial parameters

In the fusion model, the GA optimises the HMM parameters for the given input sequences, transformed by the ANN (see Fig. 5). The parameters of the HMM are re-estimated using the Baum–Welch expectation maximization algorithm. The performance of this technique in fact depends on the initial values of the HMM parameters (Smyth, 1997; Sankar, 1998; Kwong & Qianhua, 2001) and the input observation sequences. Eq. (5) also suggests this dependency. In HiMMI (Hassan & Nath, 2005) the values of the transition matrix, observation probability matrix and prior probability matrix were chosen randomly, while the input observation sequences were not transformed. Here in the fusion model, we employ the GA to obtain optimized initial values of the HMM parameters so that, after training the HMM becomes the best suited model for the observation sequences transformed by the ANN.

GA has been used in a number of studies (Fogel, 1993) to obtain an optimal solution. The strength of global searching within the searching space makes it a unique tool for any optimization problem especially for those areas where exact mathematical solution is not possible. The stock market datasets and even the transformed data in subsection 3.1.1 are continuous in behaviour, and hence it is a tedious task to find out the underlying data distribution. For the continuous HMM, the observation emission probability distribution is represented by Gaussian mixtures (Eq. (4)). Still to assume the initial values of mean and covariance matrices is a problem. For this problem the GA is the appropriate tool to find out these optimal initial values.

To optimize HMM (parameter values or architecture) researchers have proposed different approaches, while most of them adopt some heuristics. There are few studies where GA are used along with HMM but in a different approach (Chau, Kwong, Diu, & Fahrner, 1997). Here, we use the GA in a novel way to get an optimal initial parameter values for the HMM. The method is described in detail below.

There are three parameters (1. Observation emission probability matrix, 2. State transition probability matrix and 3. Prior probability matrix) in HMM which are to be optimised. If we use a canonical GA to evolve values for all the parameters, the length of the chromosome becomes very large making the overall process very complex and expensive. The alternative way is to divide the problem into three parts as is done in co-evolutionary models: only one parameter values of the HMM are optimized at a time while the other two parameter values are kept fixed. The Fig. 5 shows the process of optimising the fusion model. The steps of the optimisation are:

1. Initially choose the parameter values randomly.
2. Execute the GA to obtain initial values of the observation emission probability matrix keeping the other initial parameter values as is found in step 1.
3. Execute the GA to obtain initial values of the state transition probability matrix keeping the initial observation emission probability matrix values obtained in step 2 and the prior probability values generated in step 1.
4. Execute the GA to obtain initial values of the prior probability matrix keeping the initial observation emission probability matrix values obtained in step 2 and the state transition probability matrix values obtained in step 3.
5. If the resulting fitness value converges go to step 2.

The construction of GA algorithm is as follows.

The size of the chromosome is equal to the dimension of parameters which is to be optimized. For instance for a one dimensional Gaussian probability distribution as emission probability distribution the size of the chromosome will be 16 (for four state HMM). Again for the prior probability matrix the chromosome will be four for a 4-state HMM. The objective function is the minimization of Mean Absolute Percentage Error (MAPE) of the ANN-HMM forecast method. A portion of the training dataset is used in the GA part of the method to get optimized parameter values and calculate the output of the objective function. The steps to obtain optimized initial emission probability matrix are:

1. Initialize population.
2. Evaluate population.
3. Train the HMM by using each chromosome of the population as initial emission probability matrix and the transformed observation sequences.
4. Obtain forecast value for the validation dataset by finding the similar data vector and calculating the price difference.
5. Calculate the MAPE value for the validation dataset.
6. If termination condition is not met go to next step else exit.
7. Select parents based on the MAPE produced, for the next population.
8. Perform crossover and mutation.
9. Evaluate population.
10. Go to step 3.

3.2. Using weighted average to forecast stock price

In the HiMMI model a single match of current day's stock behaviour is extracted from historical data and then price difference is calculated (Hassan & Nath, 2005). This is done assuming that, if the stock behaviour on the mth day is found to be the same as that of current 't' day the stock behaviour on day '$t+1$' will be same to that of day '$m+1$'. However this does not give importance to recent stock market behaviour, which has profound influence on the future direction of the market. Many statistical forecasting techniques (e.g. Random walk, Moving Average, Auto Regression, Weighted Moving Average, etc.) do consider recent price fluctuations in the stock market. Thus, it is reasonable to consider incorporating such trends into our fusion model.

Here, we devise an equation such that the most recently matched day is given more weight than the matched day that is far from current day. To implement this, we need to identify a range of data vectors from the past training dataset which have produced likelihood values closer to that of current data vector (i.e., a group of days is pointed out when the stock market price fluctuation was same as that of the current day). Now for each of the matched day, the price differences between those days and "one day ahead" neighbouring days are calculated. Then the Eq. (6) is used to calculate the weighted average of these price differences. The details are explained below.

Let us assume that the training dataset is ordered in the following sequence: a, b, c, d, e, f, g, h, i, j etc, where 'j' is the current day and 'a' is from the distant past. Suppose the day 'a', 'c', 'd', 'h' and 'i' matched with that of day 'j'. Denoting the price difference between days 'x' and '$x+1$' is by 'diff_x' and assigning proportionally more weight to recent matches, a weighted average of the price differences is calculated. The following equation calculates the weighted average of the price differences. The graph in Fig. 6 shows how more weights are assigned to recent day than the distant past days.

$$\text{wd}_i = \frac{\sum_m w_m^* \text{diff}_m}{\sum_m w_m} \tag{6}$$

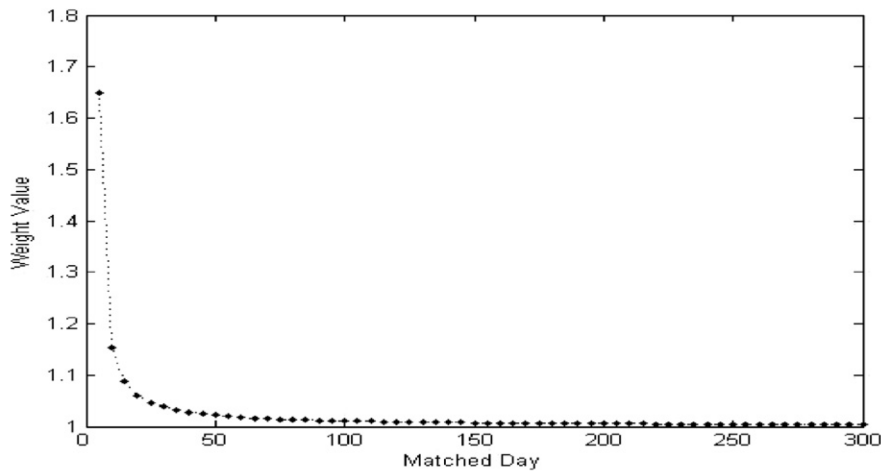

Fig. 6. The weights for the matched day (recent to far distant from current day).

where,

i index number of current day
m index number of matched day
w_m weight assigned to the day 'm' using the equation $w_m = \exp(1/i - m + 1)$
wd_i weighted average of price difference for current day 'i'
diff_m price difference between day 'm' and '$m + 1$'

Thus, a forecast value for day '$i + 1$', fp_{i+1}, is constructed as

$$\mathrm{fp}_{i+1} = p_i + \mathrm{wd}_i \tag{7}$$

where, p_i is the price on current day 'i'.

4. Experiments and results

4.1. Test data

To test the efficacy of the proposed method we have used stock prices in the IT sector: the daily stock price of Apple Computer Inc., International Business Machines Corporation (IBM) and Dell Inc., collected from www.finance.yahoo.com.

Table 1 shows the information of the training and test datasets. As in the previous study (Hassan & Nath, 2005) the four attributes: open, high, low and close price from the daily stock market are used to form the observation vector. The forecast variable here is the next day's closing price.

4.2. Experimental setup

The number of states in the HMM is set to the number of attributes in the observation vectors, as per the HiMMI. Being suitable for the continuous time series dataset left to right (Bakis model) HMM is chosen for the proposed fusion model. In the ANN section of the model we have chosen arbitrarily a three layer (an input layer, a hidden layer and an output layer) feed-forward fully connected architecture as is shown in Fig. 2. There are four neurons in the hidden layer while there must be four input neurons (due to four input variables) and four output neurons. The activation function for the hidden neurons is chosen tan-h sigmoidal function, while that of output neurons a linear function is chosen.

As mentioned in Section 3.1.1 a Gaussian normal probability distribution is chosen as emission probability distribution. Here, the parameters will be optimized with the help of GA. Initially the values for prior probabilities and state transition probabilities of the HMM are chosen equal values for each states. Then the GA is used to get optimal values for these parameters as described in Section 3.1.1. The size of the chromosomes of GA for optimizing emission probability distribution is 16 (as there are four states and four distinct variables, there should be 16 distribution functions), while the chromosome size for optimizing prior probability is 4 and that of transition probability is 16. The Table 4 (see Appendix) shows the parameter values for GA used in the model.

Table 1
Training and test data information

Stock name	Training data		Test data	
	From	To	From	To
Apple Computer Inc.	10 February 2003	10 September 2004	13 September 2004	21 January 2005
IBM Corporation	10 February 2003	10 September 2004	13 September 2004	21 January 2005
Dell Inc.	10 February 2003	10 September 2004	13 September 2004	21 January 2005

Table 2
The performance improvement of the fusion models

Stock name	Mean absolute % error (MAPE) in forecast for 91 sequential test dataset		
	HiMMI	The proposed fusion model (ANN-GA-HMM-Interpolation)	The proposed fusion model with weighted average (ANN-GA-HMM-WA)
Apple Computer Inc.	2.8373	2.16492	1.9247
IBM Corporation	1.2186	1.0555	0.84871
Dell Inc.	1.01173	0.84463	0.699246

Table 3
Forecast accuracy comparison with the ARIMA

Stock name	Mean absolute % error (MAPE) in forecast for the 91 sequential test dataset	
	The proposed fusion model with weighted average	ARIMA
Apple Computer Inc.	1.9247	1.8009
IBM Corporation	0.84871	0.9723
Dell Inc.	0.699246	0.66035

4.3. Performance metric

The performance of the fusion method is measured in terms of Mean Absolute Percentage Error (MAPE). It is calculated by first taking the absolute deviation between the actual value and the forecast value. Then the total of the ratio of deviation value with its actual value is calculated. The percentage of the average of this total ratio is the mean absolute percentage error. The following equation shows the process to calculate the MAPE.

$$\text{Mean Absolute Percentage Error(MAPE)} = \frac{\sum_{i=1}^{r}\left(\frac{\text{abs}(y_i - p_i)}{y_i}\right)}{r} \times 100\% \tag{8}$$

where:

r total number of test data sequences
y_i actual stock price on day i
p_i forecast stock price on day i

4.4. Result

Table 2 presents the experiment results for each of the three stocks considered. The second column lists results obtained using the HiMMI introduced in Hassan and Nath (2005). The third and fourth columns list the result for the fusion model using interpolation and weighted-average respectively. It is interesting to note the improved performance of the fusion-weighted average model. A comparative result in terms of MAPE with the ARIMA (p,q,r) model is given in Table 3. To use the ARIMA model for the dataset described here, we first analysed the series to obtain the order of p, q, and r, (p – the autoregressive parameter, q – the moving average parameter, r – the number of differencing passes). As a result, the best possible ARIMA (p,q,r) model is built for each of the stock prices used in this experiment.

5. Discussion

In this study, a forecast tool integrating the ANN and GA with HMM followed by a weighted average is proposed for one day ahead forecasts of stock prices. To obtain a forecast value we input the daily prices (open, high, low and close) to this tool. This tool then automatically adapts with the training datasets and makes forecast, given the current day's stock prices. The same data is used to train another well established statistical tool ARIMA. Then we produce forecast values for the test data using this ARIMA model. In doing so, we had to analyse the training dataset to choose the values of p, q, and r. As we know, using the variable of interest an autocorrelation coefficient function is generated and partial autocorrelation coefficient function is also generated. Then, analysing both of these functions together a decision is made about the values of the p, q, and r. These are done to examine whether there is any seasonality in the series. In fact, most statistical tools for forecasting are constrained with the underlying seasonality and non-stationarity (if any) (Tambi, 2005). In contrast, the proposed fusion model does not require any prior analysis of the dataset while using the statistical theories (as HMM provides a probabilistic framework for modelling a time series of multivariate observations).

In the first phase of the proposed fusion model, the ANN transforms the actual observations sequences to a set of independent values. Fig. 4 shows the transformed and the actual observation sequences. It is hard to differentiate the four sequences before transformation (see Fig. 4(a)). However, in the transformed figure, Fig. 4(b), the four individual sequences can be identified easily. Hence, it is more likely that, the transformed observation sequences might play a better role in predicting the value of variable of interest. Simultaneously, the GA in Section 3.1.2 optimizes the initial parameters to be well fitted with the transformed observation sequences. The transformed independent values along with the optimized initial parameters help the HMM to find out the similar stock behavioural data from the historical dataset more efficiently and accurately. Table 2 (second and third columns) shows the forecast accuracy improvement over the HiMMI (Hidden Markov Model followed by interpolation) model after integrating the ANN and GA with the HMM. In Section

3.1.2, Eq. (6) gives more weight to the recent matched stock data than to the far distant matched data. Fig. 6 shows how more importance is given to the recent matched data. Applying the equation of weighted average to the integrated HMM-ANN-GA (fusion) model, forecast accuracy of the tool has been boosted up significantly (see Table 2 column 4).

Looking at Table 3, we find that, the accuracy of the forecast value of the proposed fusion model is as good as that of the ARIMA model. The forecast accuracies of the proposed model, for the Apple and Dell are similar to that of ARIMA model. While the accuracy of the proposed model for the IBM is slightly better than that of ARIMA.

6. Conclusion

In this paper we described a novel time series forecasting tool. The fusion model combines a Hidden Markov Model(HMM), Artificial Neural Networks (ANN) and Genetic Algorithms(GA) to forecast financial market behaviour. As a result we find that the performance of the fusion tool is better than that of the basic model (Hassan & Nath, 2005) where only a single HMM is used in a novel approach to forecast stock price. To evaluate the efficacy of the fusion model we compare the obtained forecast accuracy with that of a popular statistical forecasting tool. The comparison shows the forecasting ability of the fusion model is as good as that of ARIMA model. Additionally, the proposed fusion model can be used without analysing the dataset prior to the forecast. That is users do not have to have carry out seasonality tests, regime analysis or cycle analysis before adopting the model. However, in the proposed fusion method, to simplify the implementation we have chosen the number of states as the number attributes in the observation vectors. This may not be suitable for some instances. To solve this, we plan to employ another GA to find the best HMM architecture for a given dataset.

Appendix

See Table 4.

Table 4
The GA parameters

Parameter for optimization in the HMM	GA parameter name	GA parameter values
Emission probability distribution function	Chromosome size	16
	Population type	Double
	Population size	20
	Elite parent selection	2
	Crossover fraction	0.8
	Migration fraction	0.2
	Generations	100
	Fitness limit	-infinity
	Initial population	Random
	Fitness scaling	Rank scaling
	Selection	Stochastic uniform
Prior probability matrix	Chromosome size	4
	Population type	Double
	Population size	20
	Elite parent selection	2
	Crossover fraction	0.8
	Migration fraction	0.2
	Generations	100
	Fitness limit	-infinity
	Initial population	Random
	Fitness scaling	Rank scaling
	Selection	Stochastic uniform
Transition probability matrix	Chromosome size	16
	Population type	Double
	Population size	20
	Elite parent selection	2
	Crossover fraction	0.8
	Migration fraction	0.2
	Generations	100
	Fitness limit	-infinity
	Initial population	Random
	Fitness scaling	Rank scaling
	Selection	Stochastic uniform

References

Abraham, A., Nath, B., & Mohanthi, P. K. (2001). Hybrid intelligent systems for stock market analysis. In Vassil N. Alexandrov et al. (Eds.), *Computational science*. Germany, San Fransisco, USA: Springer-Verlag, pp. 337–345.

Bengio, Y., Mori, R. D., & Kompe, R. (1992). Global optimization of a neural network-hidden markov model hybrid. *IEEE Transactions on Neural Networks, 3*(2), 252–259.

Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis: forecasting and control*. San Fransisco, CA: Holden-Day.

Chau, C. W., Kwong, S., Diu, C. K., & Fahrner, W. R. (1997). Optimization of HMM by a genetic algorithm. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1727–1730.

Cheng, B., & Titterington, D. M. (1994). Neural networks: a review from statistical perspective. *Statistical Science, 9*(1), 2–54.

Cheung, L. W. K. (2004). Use of runs statistics for pattern recognition in genomic DNA sequences. *Journal of Computational Biology, 11*(1), 107–124.

Chiang, W.-C., Urban, T. L., & Baldridge, G. W. (1996). A neural network approach to mutual fund net asset value forecasting. *Omega International Journal of Management Science, 24*(2), 205–215.

Choi, J. H., Lee, M. K., & Rhee, M. W. (1995). Trading S&P 500 stock index futures using a neural network. In *Proceedings of the 3rd annual international conference on artificial intelligence applications on wall street* (pp. 63–72). New York.

Coast, D. A., Stern, R. M., Cano, G. G., & Briller, S. A. (1990). An approach to cardiac arrhythmia analysis using hidden Markov mosels. *IEEE Transactions on Biomedical Engineering, 37*(9), 826–836.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. *Econometrica, 50*, 987–1008.

Fogel, D. B. (1993). Using evolutionary programming to create neural networks that are capable of playing tic-tac-toe. In *Proceedings of IEEE international conference on neural networks* (pp. 875–880).

Hassan, M. R., & Nath, B. (2005). Stock market forecasting using hidden markov model: a new approach. In *Proceedings of 5th international conference on intelligent system design and application* (pp. 192–196). Poland.

Henry, M. K. Mok (1993). Causality of interest rate, exchange rate and stock prices at stock market open and close in Hong Kong. *Asia Pacific Journal of Management, 10*(2), 123–143.

Kim, K.-J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications, 19*, 125–132.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. In *Proceeding of the international joint conference on neural networks (IJCNN)* (Vol. 1, pp. 1–6.) San Diego.

Kolarik, T., & Rudorfer, G. (1994). Time series forecasting using neural networks. *Time Series and Neural Networks*, 86–92.

Kwong, S., & Qianhua, He (2001). The use of adaptive frame for speech recognition. *EURASIP Journal on Applied Signal Processing, 2*, 82–88.

Lawrence, R. (1997). Using Neural Networks to Forecast Stock Market Prices. Available from www.cs.uiowa.edu/~rlawrenc/research/Papers/nn.pdf.

Nadas, A., Nahamoo, D., & Picheny, M. A. (1988). On model-robust training method for speech recognition. *IEEE Transactions on Acoustic, Speech and Signal Processing, 77*, 257–285.

Pankratz, A. (1983). *Forecasting with Univariate Box-Jenkins models: Concepts and Cases*. New York: John-Wiley.

Rabinar, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE, 77*, 257–285.

Raymond, Y. C. Tse (1997). An application of the ARIMA model to real-estate prices in Hong Kong. *Journal of Property Finance, 8*(2), 152–163.

Ripley, B. D. (1993). Statistical aspects of neural networks. In O. E. Brandorff-Nielsen, J. L. Jensen, & W. S. Kendall (Eds.), *Networks and chaos-statistical and probabilistic aspects* (pp. 40–123). London: Chapman and Hall.

Romahi, Y., & Shen, Q. (2000). Dynamic financial forecasting with automatically induced fuzzy associations. In *Proceedings of the 9th international conference on fuzzy systems* (pp. 493–498).

Sankar, A. (1998). Experiments with Gaussian Merging-splitting algorithm for HMM training for speech recognition. In *Proceedings of DARPA speech recognition workshop*. Avialable from www.nist.gov/speech/publications/darpa98/html/am10/am10.htm.

Smyth, P. (1997). Clustering sequences with hidden Markov models. In G. Tesauro, D. Touretzky, & T. Leen (Eds.). *Advances in neural information processing systems* (Vol. 9, pp. 648–654). Cambridge, MA: MIT Press.

Tambi, M.K. (2005). Forecasting exchange rate a uni-variate out of sample approach (Box–Jenkins Methodology). 0506005 International Finance–Economics Working Paper Archive.

Tong, H., & Lim, K. S. (1980). Threshold autoregressive, limit cycles and cyclical data. *Journal of the Royal Statistical Society Series B, 42*(3), 245–292.

Trafalis, T. B. (1999). Artificial neural networks applied to financial forecasting. In C. H. Dagi Dagli, A. L. Buczak, J. Ghosh, M. J. Embrechts, & O. Ersoy (Eds.), *Smart engineering systems: neural networks, fuzzy logic, data mining, and evolutionary programming. Proceedings of the artificial neural networks in engineering conference (ANNIE'99)* (pp. 1049–1054). New York: ASME Press.

White, H. (1988). Economic prediction using neural networks: the case of IBM daily stock returns. In *Proceedings of the second IEEE annual conference on neural networks, II* (pp. 451–458).

White, H. (1989). Learning in artificial neural networks: a statistical perspective. *Neural Computation, 1*, 425–464.

Xie, H., Andreae, P., Zhang, M., & Warren, P. (2004). Learning models for english speech recognition. In *Proceedings of the 27th conference on Australasian computer science* (pp. 323–329).

Yang, Y., & Liu, R. (2001). Multivariate time series prediction based on neural networks applied to stock market. *IEEE International Conference on Systems, Man, and Cybernetics, 4*, 2680.

Zhang, G., Patuwo, B. E., & Hu, M. H. (1998). Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting, 14*, 35–62.