Research Article

Modeling and Computing of Stock Index Forecasting Based on Neural Network and Markov Chain

Yonghui Dai,¹ Dongmei Han,^{1,2} and Weihui Dai³

¹ School of Information Management and Engineering, Shanghai University of Finance and Economics, 777 Guoding Road, Shanghai 200433, China

² Shanghai Financial Information Technology Key Research Laboratory, 777 Guoding Road, Shanghai 200433, China

³ School of Management, Fudan University, 220 Handan Road, Shanghai 200433, China

Correspondence should be addressed to Weihui Dai; whdai@fudan.edu.cn

Received 30 August 2013; Accepted 10 March 2014; Published 23 March 2014

Academic Editors: J. Shu and F. Yu

Copyright © 2014 Yonghui Dai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The stock index reflects the fluctuation of the stock market. For a long time, there have been a lot of researches on the forecast of stock index. However, the traditional method is limited to achieving an ideal precision in the dynamic market due to the influences of many factors such as the economic situation, policy changes, and emergency events. Therefore, the approach based on adaptive modeling and conditional probability transfer causes the new attention of researchers. This paper presents a new forecast method by the combination of improved back-propagation (BP) neural network and Markov chain, as well as its modeling and computing technology. This method includes initial forecasting by improved BP neural network, division of Markov state region, computing of the state transition probability matrix, and the prediction adjustment. Results of the empirical study show that this method can achieve high accuracy in the stock index prediction, and it could provide a good reference for the investment in stock market.

1. Introduction

The stock market is filled with the coexistence of high-risk and high-yield characteristics. As a barometer of the stock market, the stock index is an important reference for investors to make investment strategies. However, the stock price index is influenced by many factors such as the economic situation, policy changes, and emergency. Although faced with complicated challenges, the forecast of stock index has still attracted the attention of many industrial experts and scholars. Lendasse et al. used a nonlinear time series model to forecast the tendency of the Bel 20 stock market index [1]. Lee et al. forecasted Korean Stock Price Index (KOSPI) by three forecasting models including back-propagation neural network model (BPNN), Bayesian Chiao's model (BC), and the seasonal autoregressive integrated moving average model (SARIMA) [2]. Fan and Gao proposed "Grey Neural Network model (GNNM(1, N))" and argued that the combined model could improve the prediction accuracy and reduce the computation [3].

Up to now, stock prediction has still been a hot topic. In this field, there have been a lot of methods, such as artificial neural networks [4, 5], time series model [6, 7], decision trees [8], Bayesian belief networks [9], evolutionary algorithms [10], fuzzy sets [11], and Markov model [12–14]. However, the single method is usually limited to achieving an ideal precision in the dynamic market due to complicated influencing factors. In recent years, some new hybrid models have shown the potential superiorities [15–17]. Especially, the approach based on adaptive modeling and conditional probability transfer may be suitable for matching the problem's characteristics.

In order to explore the new solution for improving the forecast precision, this paper presented a new method based on BP neural network and Markov chain, studied its modeling and computing technology with the data of Chinese Growth Enterprise Market, and hereafter conducted an empirical analysis of the prediction results. This paper is arranged as the following five sections: Section 1 is the introduction of research background and the most related

literature; Section 2 expounds the methodology and technology as well as the combined model based on BP neural network and Markov chain; Section 3 discusses the modeling and computing technology of the presented method; Section 4 is the empirical analysis of prediction results; and the conclusion and discussion are finally in Section 5.

2. Methodology and Technology

2.1. BP Neural Network (BPNN). BPNN is a one-way propagation of BP algorithm based on multilayer network. It is based on gradient descent method which minimizes the total of the squared errors between the actual and the desired output values. The structure of three layers BPNN includes the input layer, the hidden layer, and the output layer. The BP learning algorithm of three layers can be described as follows [18, 19].

Step 1. Initialize all the values of $w_{ij}(t)$, $w_{jk}(t)$, $\theta_j(t)$, $\theta_k(t)$ to small random values within the range $[-1, 1]$, where $w_{ij}(t)$ means the connection weights between neurons i in the input layers and neurons j in the hidden layers during the t th learning process, $w_{jk}(t)$ represents the connection weights between neurons j in the hidden layers and neurons k in the output layers during the t th learning process, $\theta_j(t)$ means the threshold value in the hidden layers, and $\theta_k(t)$ means the threshold value.

Step 2. Select sample data and then apply the input vector $X(i) = (x_1, x_2, \dots, x_n)$ and desired output vector $D(i) = (d_1, d_2, \dots, d_n)$.

Step 3. Compute the outputs y'_j in every hidden layer, and compute the outputs y_k in output layer; here, $f(\cdot) = 1/(1 + e^{-x})$ or $f(\cdot) = (1 - e^{-x})/(1 + e^{-x})$ is adapted activation function:

$$\begin{aligned} y'_j &= f\left(\sum_{i=1}^n w_{ij}(t) \times x_i - \theta_j(t)\right), \\ y_k &= f\left(\sum_{j=1}^p w_{jk}(t) \times y'_j - \theta_k(t)\right). \end{aligned} \quad (1)$$

Step 4. Calculate the error terms $\delta_o(k)$ for the output nodes:

$$\delta_o(k) = y_k(1 - y_k)(d_k - y_k), \quad (2)$$

where d_k represents desired output.

Step 5. Calculate the error terms $\delta_j(k)$ for the hidden nodes:

$$\delta_j(k) = y'_j(1 - y'_j)\left(\sum_{k=1}^n w_{jk}\delta_o(k)\right). \quad (3)$$

Step 6. Update weights on the output layer:

$$w_{jk}(n) = w_{jk}(n-1) + \Delta w_{jk}(n), \quad (4)$$

where

$$\begin{aligned} \Delta w_{jk}(n) &= -\eta \frac{\partial e}{\partial w_{jk}} + \alpha \Delta w_{jk}(n-1) \\ &= \eta \delta_o(k) y_k + \alpha \Delta w_{jk}(n-1). \end{aligned} \quad (5)$$

Step 7. Update weights on the hidden layer:

$$w_{ij}(n) = w_{ij}(n-1) + \Delta w_{ij}(n), \quad (6)$$

where

$$\begin{aligned} \Delta w_{ij}(n) &= -\eta \frac{\partial e}{\partial w_{ij}} + \alpha \Delta w_{ij}(n-1) \\ &= \eta \delta_j(k) x_i + \alpha \Delta w_{ij}(n-1). \end{aligned} \quad (7)$$

Step 8. Calculate error; repeat Steps 2–8 until the error falls below a predefined threshold:

$$E_p = \frac{1}{2} \sum_{j=1}^m (d_k - y_k)^2, \quad (8)$$

where m means the number of output node.

Although BP algorithm is successful, it has some disadvantages such as lower convergence speed and easy to get into local minima points. Therefore, improved BP algorithm was applied in our study. Our improved method is based on the additional momentum and adaptive learning rate combined. The formula with the momentum factor weight adjusting is as follows:

$$\Delta w(k+1) = \text{lr} \times \text{mc} \times \frac{\partial E}{\partial w} + \text{mc} \times \Delta w(k), \quad (9)$$

where w represents network weight, k is number of training, lr is the learning rate, mc is the momentum coefficient, $0 < \text{mc} < 1$, and E is the error function.

In addition, adaptive learning rate method can be described as follows:

$$\text{lr}(k+1) = \begin{cases} \alpha \times \text{lr}(k) & \text{if } E(i+1) < E(i) \\ \beta \times \text{lr}(k) & \text{if } E(i+1) > \gamma \times E(i) \\ \text{lr}(k) & \text{otherwise,} \end{cases} \quad (10)$$

where lr is the learning rate, k is the number of training, E is the error function, $E = \sum_{i=1}^n (y_i - \hat{y}_i)^2$, y_i is actual output value, and $\hat{y}_i$ is anticipative output value; usually $\alpha = 1.05$, $\beta = 0.7$, and $\gamma = 1.04$ [20].

2.2. Markov Chain. Discrete-time Markov chain can be described as a sequence of random variables $\{X(t), t \in T\}$, where $T = \{1, 2, \dots, N\}$ and state space $S = \{1, 2, \dots, M\}$. For any time $n \geq 0$ and any state $(i_0, i_1, \dots, i_{n-1}, i, j) \in S$ and

TABLE 1: Sample data.

Days	1	2	3	4	5	6	7	8	9	10
Opening price	1044.29	1075.83	1089.77	1055.60	1063.00	1074.48	1070.73	1049.10	1021.02	1027.05
Highest price	1073.78	1091.25	1090.16	1067.03	1078.46	1086.33	1074.82	1049.26	1038.03	1031.36
Lowest price	1044.29	1075.83	1055.64	1053.18	1059.83	1068.07	1050.21	1011.40	1021.02	1020.83
Closing price	1073.78	1089.72	1059.00	1065.96	1073.87	1073.02	1052.14	1022.46	1032.76	1022.13
Trading volume	15023082	16132262	15413445	11569457	12702507	14032823	12668299	12745388	9397943	9843383
Days	11	12	13	14	15	...	55	56	57	58
Opening price	1022.56	1019.47	1029.93	1074.53	1073.31	...	1161.49	1175.63	1177.85	1158.96
Highest price	1039.65	1029.28	1069.81	1091.91	1078.53	...	1180.61	1186.29	1184.40	1178.84
Lowest price	1018.45	986.05	1029.93	1068.61	1053.33	...	1151.68	1160.02	1165.77	1131.44
Closing price	1030.14	1029.28	1068.15	1073.69	1072.93	...	1172.33	1180.66	1167.09	1132.09
Trading volume	10557897	9936156	11803687	13475017	11576013	...	13353812	16202467	15192214	15946678

Data source: the above data are from Wind information database.

positive integer step k , when this sequence of variables has the following attributes:

$$\begin{aligned}
 P \{X(n+k) = i_{n+k} \mid X(n) = i_n, \\
 X(n-1) = i_{n-1}, \dots, \\
 X(j_2) = i_{j_2}, \\
 X(j_1) = i_{j_1}\}
 \end{aligned} \quad (11)$$

$$= P \{X(n+k) = i_{n+k} \mid X(n) = i_n\},$$

$$P \{X(n+1) = j \mid X(n) = i\} = p_{ij}. \quad (12)$$

We call such stochastic variable sequence $\{X(t), t \in T\}$ Markov chains, where p_{ij} is the transition probability from state i to state j . These transition probabilities satisfied $\sum_{j \in S} p_{ij} = 1, i \in S$, and the matrix $P = (p_{ij})$ is the transition matrix of the chain. If the transition probabilities in (12) do not depend on the time parameter n , it will be called "time-homogeneous Markov chains."

Since the state space S is countable, we can label the states by integers, such as $S = \{0, 1, 2, \dots\}$. Under this label, the transition matrix can be described as follows:

$$P(n) = P(0)P_n = P(0) \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{22} & \dots & p_{2n} \\ \dots & \dots & \dots & \dots \\ p_{n1} & p_{n2} & \dots & p_{nn} \end{bmatrix}^n. \quad (13)$$

2.3. Modeling of Forecast Based on Improved BPNN and Markov Chain. The modeling process can be described as follows.

Step 1. Construct improved BPNN model.

Step 2. Initialize forecasting by using model of Step 1.

Step 3. Normalize the error of prediction. The normalized formula is as follows:

$$\hat{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \quad (14)$$

Step 4. Set Markov state zoning by normalized upper and lower thresholds.

Step 5. Divide the Markov state region by using the sample average-mean square deviation method. Five ranges are divided as follows [21]: $(-\infty, \bar{x} - \alpha_1 s)$, $(\bar{x} - \alpha_1 s, \bar{x} - \alpha_2 s)$, $(\bar{x} - \alpha_2 s, \bar{x} + \alpha_3 s)$, $(\bar{x} + \alpha_3 s, \bar{x} + \alpha_4 s)$, and $(\bar{x} + \alpha_4 s, +\infty)$, where $\bar{x}$ means average and s is sample standard deviation; usually α_1 and α_4 are range $[1.0, 1.5]$ and α_2 and α_3 are range $[0.3, 0.6]$.

Step 6. Define the initial state and calculate the state transition probability matrix.

Step 7. Markov chain test: use chi-square statistics test for Markov property.

Step 8. Forecast. Get the state vector of k step from formula (13) and forecast based on this model.

3. Modeling and Computing

3.1. Sample Data. In this paper, we select "Chinese Growth Enterprise Market Index (GEMI, 399006.SZ)" for the data set to empirical study, and then we will finish short-term Chinese GEM index price prediction based on this data set. The data set is total of 58 days, which is from 2013-5-24 to 2013-8-16 of trading data. Among them, divided into in-sample and out-of-sample, the first 41 days of data are in-sample as training data and then the data from 42 days are out-of-sample and used as prediction. Due to closing index price, the most important indicator for investment reference, our study focuses on the closing index price forecasting. The daily trading data including opening price, highest price, lowest price, closing price, and trading volume are used for modeling. The sample data of Chinese GEM index are shown in Table 1.

3.2. Modeling

3.2.1. Construct BP Neural Network Model

(i) *Definition of Layer Number.* According to Kolmogorov theorem, three layers can approach any continual function.

TABLE 2: Error of repeated training.

Number of neuron	9	10	11	12	13	14
Network error	1.8710	0.9392	0.2689	0.5924	0.3357	0.7099

Therefore, an input layer, a hidden layer, and an output layer are selected in this model.

(ii) *Activation Function and Training Target.* Here, the activation function of hidden layer neuron is tansig, the output layer neurons traditional function is purelin.

The training function is traingdx.

The end of training conditions is the mean square error of the accuracy of $E = 0.005$.

The circulation is 10000 times.

In this model, the initial learning rate is 0.1.

The initial momentum factor value is 0.9.

(iii) *Number of Neural Node.* The input layer node number is five, namely, items of opening price, highest price, lowest price, closing price, and trading volume. Meanwhile, data of “day 1” were regarded as the first input data in input layer.

In this model, the output layer node number is set to one; meanwhile, “closing price” of “day 2” was regarded as the first output data in output layer.

Numbers of hidden layer node depend on experience and repeated training, how many of the nodes depend on the network error; the number corresponding to the minimum network error in training will be chosen as the number of the hidden layer nodes.

The network errors which correspond to different number of neurons are shown in Table 2. It can be seen that this neural network has the minimum network error of 0.2689 when the neuron number is eleven. Therefore, we select eleven as the number of hidden layer nodes. The data in Table 2 indicate that network error cannot be reduced even if we continue to increase the number of hidden layer nodes.

3.2.2. Training. Training of the network is completed in MATLAB software. First, the training sample data is selected; then, the data is normalized. Normalization means to limit the data in a certain interval. Here, in order to limit training data in $[-1, 1]$, the `premnmx(.)` function is called. After normalization, start training network with a training set of 41 sample data; the learning rate is 0.1 and the momentum is 0.9. The network was in training till the Mean Squared Error (MSE) was less than 0.005. Finally, we get the ideal model after training the neural network. The dependence of MSE on epochs is shown in Figure 1.

It can be seen from Figure 1 that the network MSE reaches the expected MSE after 8078 steps of training, in which the training MSE is less than 0.005.

3.2.3. Forecast

(i) *Initial Forecasting Based on Improved BPNN.* According to trained network and sample data, we used rolling forecasting

FIGURE 1: The dependence of MSE on epochs.

FIGURE 2: Simulation of actual value and predicted value.

method to predict the closing index price. Part of the code in MATLAB software is shown in Algorithm 1.

The Chinese GEM index of daily closing price of simulation is shown in Figure 2. Both actual value and predicted value are shown when trading day from 42 days to 56 days.

(ii) Computing of Normalization

Step 1. Calculate the absolute residual rate of prediction days. The calculation formula is as follows:

$$y_i = \frac{x_a - x_p}{x_a} \times 100\%, \quad (15)$$

```

sample = xlsread('data.xls')
[m,n] = size(sample);
ts = sample(2:m,4);
tsx = sample(1:m-1,:);
TS = ts';
TSX = tsx';
[Pn1,minp1,maxp1] = prenmnmx(TS)
[Pn2,minp2,maxp2] = prenmnmx(TSX)
rand('state',0);
PR = [-1,1; -1,1; -1,1; -1,1; -1,1];
net=newff(PR,[11,1],{'tansig','purelin'},'traingdx','learnngdm','mse')
net.trainparam.show = 50;
net.trainparam.lr = 0.1;
net.trainparam.mc = 0.9;
net.trainparam.epochs = 10000;
net.trainparam.goal = 0.005;
net = init(net);
net = train(net,Pn2(:,1:41),Pn1(1:41));
% Forecast
test = Pn2(:,42:end);
Y = sim(net, test)
P = postmnmx(Y, minp1, maxp1)
e = (sample(42:m-1,4)'-P)./sample(42:m-1)
% Calculate total error of network
res = norm(e)
figure(1)
t = 42: m-1
% Draw figure
plot((1:m-1), sample(1:m-1),'-*',t, P, 'o')

```

ALGORITHM 1: Part of the code of MATLAB.

where x_a is the actual value of closing index price, x_p is the predicted value of closing index price, and y_i is the absolute residual rate of i day.

Step 2. Normalize the data set of absolute residual rate in MATLAB software; the `mapminmax(.)` function is called. The absolute residual rate and normalized results are shown in Table 3.

3.2.4. Empirical Markov Model

(i) *State Definition.* According to the normalization value of Table 3, sample average-mean square deviation was used in state classification. Usually five intervals are divided: $(-\infty, \bar{x} - \alpha_1 S)$, $(\bar{x} - \alpha_1 S, \bar{x} - \alpha_2 S)$, $(\bar{x} - \alpha_2 S, \bar{x} + \alpha_3 S)$, $(\bar{x} + \alpha_3 S, \bar{x} + \alpha_4 S)$, and $(\bar{x} + \alpha_4 S, +\infty)$, where $\bar{x}$ is average, S is sample standard deviation, α_1 and α_4 belong to range $[1.0, 1.5]$, and α_2 and α_3 belong to range $[0.3, 0.6]$.

Taking into account the fact that the data is not that much, Markov state was divided into four ranges according to $(-\infty, \bar{x} - 0.4S)$, $(\bar{x} - 0.4S, \bar{x} + 0.6S)$, $(\bar{x} + 0.6S, \bar{x} + 1.5S)$, and $(\bar{x} + 1.5S, +\infty)$; therefore, Markov state ranges are (1) $[0, 0.1926]$, (2) $(0.1926, 0.4840]$, (3) $(0.4840, 0.7462]$, and (4) $(0.7462, 1]$. Then, Markov state transition was built as shown in Table 4.

TABLE 3: Normalization of absolute residual rate.

Days	Actual value	Prediction value	Error of absolute residual rate	Normalization value
1	1148.81	1089.72	5.27%	0.6797
2	1133.16	1207.61	-4.25%	0.2457
3	1136.40	1198.35	-5.40%	0.1978
4	1095.65	1143.73	-1.75%	0.3648
5	1117.34	1043.10	8.02%	0.8214
6	1135.45	1217.55	-2.31%	0.3326
7	1187.31	1046.42	11.98%	1.0000
8	1182.54	1291.15	-7.34%	0.0985
9	1200.69	1242.76	-2.56%	0.3251
10	1173.26	1280.88	-8.95%	0.0397
11	1165.32	1223.61	-3.17%	0.2956
12	1153.45	1273.66	-8.63%	0.0478
13	1146.02	1254.19	-7.95%	0.0789
14	1151.68	1204.72	-2.76%	0.3153
15	1160.02	1266.82	-7.30%	0.1037
16	1165.77	1279.64	-9.64%	0.0000

TABLE 4: Markov state transition.

State	(1)	(2)	(3)	(4)	Total
(1)	2	3	0	0	5
(2)	3	2	0	2	7
(3)	0	1	0	0	1
(4)	1	1	0	0	2
Total	6	7	0	2	15

(ii) *Computing of State Transition Probability Matrix.* It can be seen from Table 4 that from the state (1) to (1) it has 2 times, from the state (1) to (2) it has 3 times, from the state (1) to (3) it has 0 times, and from the state (1) to (4) it has 0 times; then, the state transition probability can be calculated as follows:

$$\begin{aligned} p_{11} &= \frac{2}{5} = 0.4, & p_{12} &= \frac{3}{5} = 0.6, \\ p_{13} &= \frac{0}{5} = 0.0, & p_{14} &= \frac{0}{5} = 0.0. \end{aligned} \quad (16)$$

Similarly,

$$\begin{aligned} p_{21} &= \frac{3}{7} = 0.4286, & p_{22} &= \frac{2}{7} = 0.2857, \\ p_{23} &= \frac{0}{7} = 0.0, & p_{24} &= \frac{2}{7} = 0.2857, \\ p_{31} &= \frac{0}{1} = 0.0, & p_{32} &= \frac{1}{1} = 1.0, \\ p_{33} &= \frac{0}{1} = 0.0, & p_{34} &= \frac{0}{1} = 0.0, \\ p_{41} &= \frac{1}{2} = 0.5, & p_{42} &= \frac{1}{2} = 0.5, \\ p_{43} &= \frac{0}{2} = 0.0, & p_{44} &= \frac{0}{2} = 0.0. \end{aligned} \quad (17)$$

Thus, the state transition probability matrix p can be described as follows:

$$p = \begin{bmatrix} 0.4 & 0.6 & 0.0 & 0.0 \\ 0.4286 & 0.2857 & 0.0 & 0.2857 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.5 & 0.5 & 0.0 & 0.0 \end{bmatrix}. \quad (18)$$

The probability matrix p has the Markov property after chi-square statistics test.

(iii) *The Step State Vector of Prediction.* According to the state transition probability matrix p and the Markov forecast model, the step state vector of prediction can be calculated as follows:

$$P(n) = [0, 0, 1, 0] \times p^n. \quad (19)$$

Thus, the step state vector of prediction can be described as shown in Table 5.

TABLE 5: Probability of step state vector.

State	Step 1	Step 2	Step 3	Step 4
(1)	0.0000	0.4286	0.4367	0.4219
(2)	1.0000	0.2857	0.4816	0.4405
(3)	0.0000	0.0000	0.0000	0.0000
(4)	0.0000	0.2857	0.0816	0.1376
State	Step 5	Step 6	Step 7	Step 8
(1)	0.4264	0.4254	0.4256	0.4255
(2)	0.4478	0.4467	0.4468	0.4468
(3)	0.0000	0.0000	0.0000	0.0000
(4)	0.1258	0.1279	0.1276	0.1277
State	Step 9	Step 10	Step 11	Step 12
(1)	0.4255	0.4255	0.4255	0.4255
(2)	0.4468	0.4468	0.4468	0.4468
(3)	0.0000	0.0000	0.0000	0.0000
(4)	0.1277	0.1277	0.1277	0.1277
State	Step 13	Step 14	Step 15	Step 16
(1)	0.4255	0.4255	0.4255	0.4255
(2)	0.4468	0.4468	0.4468	0.4468
(3)	0.0000	0.0000	0.0000	0.0000
(4)	0.1277	0.1277	0.1277	0.1277

4. Empirical Analysis

According to the step state vector of prediction of Markov model, prediction result from 2013-7-25 to 2013-8-15 was shown in Table 6. Among them, V_{col6} is adjustment value and $V_{col6} = P_{max5} \wedge \bar{X}_{col4}$, where P_{max5} means the maximum probability of someday in fifth column and $\bar{X}_{col4}$ means the average of interval of fourth column.

It can be seen from column "error of absolute residual rate" in Table 6, during sixteen trading days, that most of the prediction results by this model are better than a single improved neural network prediction except during the first day and the fifth day.

5. Conclusion and Discussion

Due to the complicated influencing factors in dynamic stock market, the comprehensive method with hybrid models throws off more superiorities than a single method in the forecast of stock index. This paper presented a new method based on the combination of improved back-propagation (BP) neural network and Markov chain, which took the advantages of neural network and Markov model, and obtained the results better than that of the single improved BPNN method. This method could provide a good reference for the investment in stock market.

As an open complex adaptive system constantly affected by all kinds of emergency events and people's psychological and behavioral effects, although many scholars including the famous financial experts pointed out that the changes of stock market cannot be predicted, we had to break those traditional ideas which rely only on the financial theory models and explore new combined methods such as the TDF

TABLE 6: Prediction result.

Days	Actual value	Value of improved BPNN forecast	Markov prediction interval	Probability	Adjustment value	Error of absolute residual rate (improved BPNN)	Error of absolute residual rate (adjustment)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	1150.39	1089.72	[985.6, 1030.6] [1030.6, 1098.8] [1098.8, 1160.1] [1160.1, 1219.5]	0.0000 1.0000 0.0000 0.0000	1064.72	5.27%	7.45%
2	1158.35	1207.61	[1092.2, 1142.1] [1142.1, 1217.7] [1217.7, 1285.6] [1285.6, 1351.4]	0.4286 0.2857 0.0000 0.2857	1117.18	-4.25%	3.55%
3	1136.98	1198.35	[1083.8, 1133.3] [1133.3, 1208.3] [1208.3, 1275.7] [1275.7, 1341.0]	0.4367 0.4816 0.0000 0.0816	1170.85	-5.40%	-2.98%
4	1124.10	1143.73	[1034.4, 1081.7] [1081.7, 1153.2] [1153.2, 1217.6] [1217.6, 1279.9]	0.4219 0.4405 0.0000 0.1376	1117.48	-1.75%	0.59%
5	1134.02	1043.10	[943.4, 986.5] [986.5, 1051.8] [1051.8, 1110.5] [1110.5, 1167.3]	0.4263 0.4478 0.0000 0.1258	1019.17	8.02%	10.13%
6	1190.11	1217.55	[1101.2, 1151.5] [1151.5, 1227.7] [1227.7, 1296.2] [1296.2, 1362.5]	0.4254 0.4467 0.0000 0.1279	1189.61	-2.31%	0.04%
7	1188.85	1046.42	[946.4, 989.7] [989.7, 1055.1] [1055.1, 1114.0] [1114.0, 1171.0]	0.4256 0.4468 0.0000 0.1276	1084.58	11.98%	8.77%
8	1202.83	1291.15	[1167.8, 1221.1] [1221.1, 1301.9] [1301.9, 1374.6] [1374.6, 1444.9]	0.4255 0.4468 0.0000 0.1277	1261.51	-7.34%	-4.88%
9	1211.78	1242.76	[1124.0, 1175.4] [1175.4, 1253.1] [1253.1, 1323.1] [1323.1, 1390.7]	0.4255 0.4468 0.0000 0.1277	1214.24	-2.56%	-0.20%
10	1175.70	1280.88	[1158.5, 1211.4] [1211.4, 1291.5] [1291.5, 1363.6] [1363.6, 1433.4]	0.4255 0.4468 0.0000 0.1277	1251.48	-8.95%	-6.45%
11	1185.96	1223.61	[1106.7, 1157.2] [1157.2, 1233.8] [1233.8, 1302.7] [1302.7, 1369.3]	0.4255 0.4468 0.0000 0.1277	1195.53	-3.17%	-0.81%
12	1172.52	1273.66	[1151.9, 1204.6] [1204.6, 1284.2] [1284.2, 1355.9] [1355.9, 1425.3]	0.4255 0.4468 0.0000 0.1277	1244.43	-8.63%	-6.13%
13	1161.87	1254.19	[1134.3, 1186.1] [1186.1, 1264.6] [1264.6, 1335.2] [1335.2, 1403.5]	0.4255 0.4468 0.0000 0.1277	1225.40	-7.95%	-5.47%
14	1172.33	1204.72	[1089.6, 1139.4] [1139.4, 1214.7] [1214.7, 1282.5] [1282.5, 1348.1]	0.4255 0.4468 0.0000 0.1277	1177.07	-2.76%	-0.40%

TABLE 6: Continued.

Days	Actual value	Value of improved BPNN forecast	Markov prediction interval	Probability	Adjustment value	Error of absolute residual rate (improved BPNN)	Error of absolute residual rate (adjustment)
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
15	1180.66	1266.82	[1145.7, 1198.1] [1198.1, 1277.3] [1277.3, 1348.6] [1348.6, 1417.6]	0.4255 0.4468 0.0000 0.1277	1237.75	-7.30%	-4.84%
16	1167.09	1279.64	[1157.3, 1210.2] [1210.2, 1290.2] [1290.2, 1362.3] [1362.3, 1432.0]	0.4255 0.4468 0.0000 0.1277	1250.28	-9.64%	-7.13%

(Theory-Data-Feedback) modeling and analyzing framework [22] and the spread model of emotions and behaviors caused by emergency events [23]. We believe that the change of the stock market has also its characteristics and inherent rules, and the forecast is possible at least in the short-term prediction.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (no. 41174007), Graduate Innovation Fund Program (no. CXJJ-2013-445) of Shanghai University of Finance and Economics, and High Level Academic Research Program (no. 2012FDFRCGD02) of Financial Research Center, Fudan University, China.

References

- [1] A. Lendasse, E. de Bodt, V. Wertz, and M. Verleysen, "Non-linear financial time series forecasting-application to the Bel 20 stock market index," *European Journal of Economic and Social Systems*, vol. 14, no. 1, pp. 81-91, 2000.
- [2] K. J. Lee, A. Y. Chi, S. Yoo, and J. J. Jin, "Forecasting Korean stock price index (kospi) using back propagation neural network model, Bayesian Chiao's model, and ASRIMA model," *Academy of Information and Management Sciences Journal*, vol. 11, no. 2, pp. 53-62, 2008.
- [3] Y. Fan and F. Gao, "A stock index forecasting model based on grey relation theory and GNNM (1, N)," in *Proceedings of the International Conference in Electrics, Communication and Automatic Control Proceedings*, pp. 1625-1633, 2011.
- [4] C. W. Li and J. Zhang, "ANN-based mid-term stock forecasting," *Computer Engineering & Science*, vol. 28, no. 5, pp. 115-117, 2006.
- [5] M. Haniyas, P. Curtis, and E. Thalassinis, "Time series prediction with neural networks for the athens stock exchange indicator," *European Research Studies Journal*, vol. 15, no. 2, pp. 23-32, 2012.
- [6] W. Liu and B. Morley, "Volatility forecasting in the hang seng index using the GARCH approach," *Asia-Pacific Financial Markets*, vol. 16, no. 1, pp. 51-63, 2009.
- [7] P. Srinivasan, "Modeling and forecasting the stock market volatility of S&P 500 index using GARCH models," *The IUP Journal of Behavioral Finance*, vol. 8, no. 1, pp. 51-69, 2011.
- [8] B. B. Nair, V. P. Mohandas, and N. R. Sakthivel, "A decision tree-rough set hybrid system for stock market trend prediction," *International Journal of Computer Applications*, vol. 6, no. 9, pp. 1-6, 2010.
- [9] J. Ying, L. Kuo, and G. S. Seow, "Forecasting stock prices using a hierarchical Bayesian approach," *Journal of Forecasting*, vol. 24, no. 1, pp. 39-59, 2005.
- [10] M. A. Kaboudan, "Genetic programming prediction of stock prices," *Computational Economics*, vol. 16, no. 3, pp. 207-236, 2000.
- [11] H. Hwang and J. Oh, "Fuzzy models for predicting time series stock price index," *International Journal of Control, Automation and Systems*, vol. 8, no. 3, pp. 702-706, 2010.
- [12] A. Alizadeh and N. Nomikos, "A markov regime switching approach for hedging stock indices," *Journal of Futures Markets*, vol. 24, no. 7, pp. 649-674, 2004.
- [13] S. Li and X. Hui, "A new stock index fuzzy stochastic prediction model developed by introducing a Markov chain," *Journal of Harbin Engineering University*, vol. 32, no. 8, pp. 1086-1090, 2011.
- [14] J. Gong and C. H. Ma, "A hidden Markov chain modeling of shanghai stock index," *Finance*, vol. 2, pp. 45-49, 2012.
- [15] M. J. Kim, I. Han, and K. C. Lee, "Hybrid knowledge integration using the fuzzy genetic algorithm: prediction of the Korea stock price index," *Intelligent Systems in Accounting, Finance and Management*, vol. 12, no. 1, pp. 43-60, 2004.
- [16] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497-505, 2005.
- [17] M. R. Hassan, B. Nath, and M. Kirley, "A fusion model of HMM, ANN and GA for stock market forecasting," *Expert Systems with Applications*, vol. 33, no. 1, pp. 171-180, 2007.
- [18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.
- [19] J. P. Yang, *The Research of Improved BP Algorithm Based on Self-Adaptive Learning Rate*, Tianjin University, Tianjin, China, 2008.
- [20] G. L. Su and F. P. Deng, "On the improving back propagation algorithms of the neural networks based on MATLAB language: a review," *Bulletin of Science and Technology*, vol. 19, no. 2, pp. 130-135, 2003.

- [21] M. M. He, *The Application on Some Economic Prediction With Markov Chain Model*, Harbin Institute of Technology, Harbin, China, 2008.
- [22] W. H. Dai, "The public cognitive mechanism of emotion on city emergency events and coping strategy," *Urban Management*, no. 1, pp. 34–37, 2014.
- [23] W. H. Dai, X. Q. Wan, and X. Y. Liu, "Emergency event: internet spread, psychological impacts and emergency management," *Journal of Computers*, vol. 6, no. 8, pp. 1748–1755, 2011.