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

Yonghui Dai, Dongmei Han, and Weihui Dai

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
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 \( X(i) = (x_1, x_2, \ldots, x_n) \) and desired output vector \( D(i) = (d_1, d_2, \ldots, d_m) \).

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

\[
y_j' = f \left( \sum_{i=1}^{n} w_{ij}(t) \times x_i - \theta_j(t) \right),
\]

\[
y_k = f \left( \sum_{j=1}^{p} w_{jk}(t) \times y_j - \theta_k(t) \right).
\]

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

\[
\delta_o(k) = y_k \left( 1 - y_k \right) \left( d_k - y_k \right),
\]

where \( d_k \) represents desired output.

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

\[
\delta_j(k) = y'_j \left( 1 - y'_j \right) \left( \sum_{i=1}^{n} w_{jk} \delta_o(k) \right).
\]

**Step 6.** Update weights on the output layer:

\[
w_{jk}(n) = w_{jk}(n - 1) + \Delta w_{jk}(n),
\]

where

\[
\Delta w_{jk}(n) = -\eta \frac{\partial e}{\partial w_{jk}} + \alpha \Delta w_{jk}(n - 1)
\]

\[
= \eta \delta_o(k) y_k + \alpha \Delta w_{jk}(n - 1).
\]

**Step 7.** Update weights on the hidden layer:

\[
w_{ij}(n) = w_{ij}(n - 1) + \Delta w_{ij}(n),
\]

where

\[
\Delta w_{ij}(n) = -\eta \frac{\partial e}{\partial w_{ij}} + \alpha \Delta w_{ij}(n - 1)
\]

\[
= \eta \delta_j(k) x_i + \alpha \Delta w_{ij}(n - 1).
\]

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

\[
E_p = \frac{1}{2} \sum_{j=1}^{m} (d_k - y_k)^2,
\]

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:

\[
\Delta w(k + 1) = lr \times mc \times \frac{\partial E}{\partial w} + mc \times \Delta w(k),
\]

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

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

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

where \( \text{lr} \) is the learning rate, \( k \) is the number of training, \( E \) is the error function, \( E = \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2 \), \( y_i \) is actual output value, and \( \tilde{y}_i \) 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 \( T = \{1, 2, \ldots, N\} \) and state space \( S = \{1, 2, \ldots, M\} \). For any time \( n \geq 0 \) and any state \( \{i_0, i_1, \ldots, i_{n-1}, i, j\} \in S \) and
positive integer step \(k\), when this sequence of variables has the following attributes:

\[
P \{ X(n+k) = i_{n+k} | X(n) = i_n \},
\]

\[
X(n-1) = i_{n-1}, \ldots ,
\]

\[
X(j_2) = i_{j_2} , \quad (11)
\]

\[
X(j_1) = i_{j_1} \}
\]

\[
= P \{ X(n+k) = i_{n+k} X(n) = i_n \},
\]

\[
P \{ X(n+1) = jX(n) = i \} = p_{ij} . \quad (12)
\]

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 \(S = \{0, 1, 2, \ldots \}\). Under this label, the transition matrix can be described as follows:

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

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]: \((-\infty, \bar{x} - \alpha_1 s), (\bar{x} - \alpha_1 s, \bar{x} - \alpha_2 s), (\bar{x} - \alpha_2 s, \bar{x} + \alpha_3 s), (\bar{x} + \alpha_3 s, \bar{x} + \alpha_4 s), (\bar{x} + \alpha_4 s, +\infty)\), where \(\bar{x}\) means average and \(s\) is sample standard deviation; usually \(\alpha_1\) and \(\alpha_4\) are range \([1.0, 1.5]\) and \(\alpha_2\) and \(\alpha_3\) 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.
(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 traindx.

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.

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($\cdot$) 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 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:

\[
y_i = \frac{x_a - x_p}{x_a} \times 100\%
\]  

(15)
TheScientificWorldJournal 5

```matlab
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] = premnmx(TS)
[Pn2, minp2, maxp2] = premnmx(TSX)
rand('state',0);
PR=[-1,1;-1,1;-1,1;-1,1];
net=newff(PR,
[11,1],
{tansig,'purelin'},
'traingdx','learngdm','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));
Y=sim(net,test)
P=postmnmx(Y,minp1,maxp1)
e=(sample(42:m-1,4)'-P)./sample(42:m-1)
res = norm(e)
figure(1)
t=42:m-1
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, \( \alpha_1 \) and \( \alpha_4 \) belong to range \([1.0, 1.5]\), and \( \alpha_2 \) and \( \alpha_3 \) 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>
<thead>
<tr>
<th>Days</th>
<th>Actual value</th>
<th>Prediction value</th>
<th>Error of absolute residual rate</th>
<th>Normalization value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1148.81</td>
<td>1089.72</td>
<td>5.27%</td>
<td>0.6797</td>
</tr>
<tr>
<td>2</td>
<td>1133.16</td>
<td>1207.61</td>
<td>-4.25%</td>
<td>0.2457</td>
</tr>
<tr>
<td>3</td>
<td>1136.40</td>
<td>1198.35</td>
<td>-5.40%</td>
<td>0.1978</td>
</tr>
<tr>
<td>4</td>
<td>1095.65</td>
<td>1143.73</td>
<td>-1.75%</td>
<td>0.3648</td>
</tr>
<tr>
<td>5</td>
<td>1117.34</td>
<td>1043.10</td>
<td>8.02%</td>
<td>0.8214</td>
</tr>
<tr>
<td>6</td>
<td>1135.45</td>
<td>1217.55</td>
<td>-2.31%</td>
<td>0.3326</td>
</tr>
<tr>
<td>7</td>
<td>1187.31</td>
<td>1046.42</td>
<td>11.98%</td>
<td>1.0000</td>
</tr>
<tr>
<td>8</td>
<td>1182.54</td>
<td>1291.15</td>
<td>-7.34%</td>
<td>0.0985</td>
</tr>
<tr>
<td>9</td>
<td>1200.69</td>
<td>1242.76</td>
<td>-2.56%</td>
<td>0.3251</td>
</tr>
<tr>
<td>10</td>
<td>1173.26</td>
<td>1280.88</td>
<td>-8.95%</td>
<td>0.0397</td>
</tr>
<tr>
<td>11</td>
<td>1165.32</td>
<td>1223.61</td>
<td>-3.17%</td>
<td>0.2956</td>
</tr>
<tr>
<td>12</td>
<td>1153.45</td>
<td>1273.66</td>
<td>-8.63%</td>
<td>0.0478</td>
</tr>
<tr>
<td>13</td>
<td>1146.02</td>
<td>1254.19</td>
<td>-7.95%</td>
<td>0.0789</td>
</tr>
<tr>
<td>14</td>
<td>1151.68</td>
<td>1204.72</td>
<td>-2.76%</td>
<td>0.3153</td>
</tr>
<tr>
<td>15</td>
<td>1160.02</td>
<td>1266.82</td>
<td>-7.30%</td>
<td>0.1037</td>
</tr>
<tr>
<td>16</td>
<td>1165.77</td>
<td>1279.64</td>
<td>-9.64%</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
Table 4: Markov state transition.

<table>
<thead>
<tr>
<th>State</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>(2)</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>(3)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(4)</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>6</td>
<td>7</td>
<td>0</td>
<td>2</td>
<td>15</td>
</tr>
</tbody>
</table>

(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:

\[
p_{11} = \frac{2}{5} = 0.4, \quad p_{12} = \frac{3}{5} = 0.6, \quad p_{13} = \frac{0}{5} = 0.0, \quad p_{14} = \frac{0}{5} = 0.0.
\]

Similarly,

\[
p_{21} = \frac{3}{7} = 0.4286, \quad p_{22} = \frac{2}{7} = 0.2857, \quad p_{23} = \frac{0}{7} = 0.0, \quad p_{24} = \frac{2}{7} = 0.2857,
\]

\[
p_{31} = \frac{0}{1} = 0.0, \quad p_{32} = \frac{1}{1} = 1.0, \quad p_{33} = \frac{0}{1} = 0.0, \quad p_{34} = \frac{0}{1} = 0.0,
\]

\[
p_{41} = \frac{1}{2} = 0.5, \quad p_{42} = \frac{1}{2} = 0.5, \quad p_{43} = \frac{0}{2} = 0.0, \quad p_{44} = \frac{0}{2} = 0.0.
\]

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

\[
p = \begin{bmatrix}
0.4 & 0.6 & 0.0 & 0.0 \\
0.4286 & 0.2857 & 0.0 & 0.2857 \\
0.0 & 1.0 & 0.0 & 0.0 \\
0.5 & 0.5 & 0.0 & 0.0
\end{bmatrix}.
\]

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:

\[
\mathbf{P}(n) = [0, 0, 1, 0] \times p^n.
\]

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

Table 5: Probability of step state vector.

<table>
<thead>
<tr>
<th>State</th>
<th>Step 1</th>
<th>Step 2</th>
<th>Step 3</th>
<th>Step 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>0.0000</td>
<td>0.4286</td>
<td>0.4367</td>
<td>0.4219</td>
</tr>
<tr>
<td>(2)</td>
<td>1.0000</td>
<td>0.2857</td>
<td>0.4816</td>
<td>0.4405</td>
</tr>
<tr>
<td>(3)</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>(4)</td>
<td>0.0000</td>
<td>0.2857</td>
<td>0.0816</td>
<td>0.1376</td>
</tr>
</tbody>
</table>

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 \( V_{col6} = P_{max5} \land \overline{X}_{col4} \), where \( P_{max5} \) means the maximum probability of someday in fifth column and \( \overline{X}_{col4} \) 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.

<table>
<thead>
<tr>
<th>Days</th>
<th>Actual value</th>
<th>Value of improved BPNN forecast</th>
<th>Markov prediction interval</th>
<th>Probability</th>
<th>Adjustment value</th>
<th>Error of absolute residual rate (improved BPNN)</th>
<th>Error of absolute residual rate (adjustment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1150.39</td>
<td>1089.72</td>
<td>[985.6, 1030.6]</td>
<td>0.0000</td>
<td>1064.72</td>
<td>5.27%</td>
<td>7.45%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1030.6, 1098.8]</td>
<td>1.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1098.8, 1160.1]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1160.1, 1219.5]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1158.35</td>
<td>1207.61</td>
<td>[1092.2, 1142.1]</td>
<td>0.4286</td>
<td>1117.18</td>
<td>−4.25%</td>
<td>3.55%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1142.1, 1277.7]</td>
<td>0.2857</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1217.7, 1285.6]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1285.6, 1351.4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1136.98</td>
<td>1198.35</td>
<td>[1083.8, 1133.3]</td>
<td>0.4367</td>
<td>1170.85</td>
<td>−5.40%</td>
<td>−2.98%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1133.3, 1208.3]</td>
<td>0.4816</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1208.3, 1275.7]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1275.7, 1341.0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1124.10</td>
<td>1143.73</td>
<td>[1034.4, 1081.7]</td>
<td>0.4219</td>
<td>1117.48</td>
<td>−1.75%</td>
<td>0.59%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1081.7, 1153.2]</td>
<td>0.4405</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1153.2, 1217.6]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1217.6, 1279.9]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1134.02</td>
<td>1043.10</td>
<td>[943.4, 986.5]</td>
<td>0.4263</td>
<td>1019.17</td>
<td>8.02%</td>
<td>10.13%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[986.5, 1051.8]</td>
<td>0.4478</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1051.8, 1110.5]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1110.5, 1167.3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1190.11</td>
<td>1217.55</td>
<td>[1011.2, 1151.5]</td>
<td>0.4254</td>
<td>1189.61</td>
<td>−2.31%</td>
<td>0.04%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1151.5, 1227.7]</td>
<td>0.4467</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1227.7, 1296.2]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1296.2, 1362.5]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1188.85</td>
<td>1046.42</td>
<td>[946.4, 989.7]</td>
<td>0.4256</td>
<td>1084.58</td>
<td>11.98%</td>
<td>8.77%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[989.7, 1055.1]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1055.1, 1114.0]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1114.0, 1171.0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1202.83</td>
<td>1291.15</td>
<td>[1167.8, 1221.1]</td>
<td>0.4255</td>
<td>1261.51</td>
<td>−7.34%</td>
<td>−4.88%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1221.1, 1301.9]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1301.9, 1374.6]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1374.6, 1444.9]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1211.78</td>
<td>1242.76</td>
<td>[1124.0, 1175.4]</td>
<td>0.4255</td>
<td>1214.24</td>
<td>−2.56%</td>
<td>−0.20%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1175.4, 1253.1]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1253.1, 1323.1]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1323.1, 1390.7]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1175.70</td>
<td>1280.88</td>
<td>[1158.5, 1211.4]</td>
<td>0.4255</td>
<td>1251.48</td>
<td>−8.95%</td>
<td>−6.45%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1211.4, 1291.5]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1291.5, 1363.6]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1363.6, 1433.4]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1185.96</td>
<td>1223.61</td>
<td>[1106.7, 1157.2]</td>
<td>0.4255</td>
<td>1195.53</td>
<td>−3.17%</td>
<td>−0.81%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1157.2, 1233.8]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1233.8, 1302.7]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1302.7, 1369.3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1172.52</td>
<td>1273.66</td>
<td>[1151.9, 1204.6]</td>
<td>0.4255</td>
<td>1244.43</td>
<td>−8.63%</td>
<td>−6.13%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1204.6, 1284.2]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1284.2, 1355.9]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1355.9, 1425.3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1161.87</td>
<td>1254.19</td>
<td>[1134.3, 1186.1]</td>
<td>0.4255</td>
<td>1225.40</td>
<td>−7.95%</td>
<td>−5.47%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1186.1, 1264.6]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1264.6, 1335.2]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1335.2, 1403.5]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1172.33</td>
<td>1204.72</td>
<td>[1089.6, 1139.4]</td>
<td>0.4255</td>
<td>1177.07</td>
<td>−2.76%</td>
<td>−0.40%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1139.4, 1214.7]</td>
<td>0.4468</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1214.7, 1282.5]</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1282.5, 1348.1]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6: Continued.

<table>
<thead>
<tr>
<th>Days</th>
<th>Actual value</th>
<th>Value of improved BPNN forecast</th>
<th>Markov prediction interval</th>
<th>Probability (improved BPNN)</th>
<th>Adjustment value</th>
<th>Error of absolute residual rate (improved BPNN)</th>
<th>Error of absolute residual rate (adjustment)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1180.66</td>
<td>1266.82</td>
<td>[1145.7, 1198.1]</td>
<td>0.4255</td>
<td>1237.75</td>
<td>−7.30%</td>
<td>−4.84%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1198.1, 1277.3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1277.3, 1348.6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1348.6, 1417.6]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1167.09</td>
<td>1279.64</td>
<td>[1157.3, 1210.2]</td>
<td>0.4255</td>
<td>1250.28</td>
<td>−9.64%</td>
<td>−7.13%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1210.2, 1290.2]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1290.2, 1362.3]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[1362.3, 1432.0]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(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


