Predicting the Rise and Fall of Stock Prices based on the modified BP_AdaBoost

Lu Li¹, Shizhan Xu¹, Yanhong Liu¹ and Xiaorong Yang¹,*
¹Department of statistics & Mathematics, Zhejiang Gongshang University
Hangzhou, 310016, China
*Corresponding author’s e-mail: xryang@zjgsu.edu.cn

Abstract. In the big data era, the studies of the quantitative stock selection strategy based on machine learning are becoming more and more popular. Most of existing studies focus on short-term strategies, and few on the medium-term or long-term strategies. Moreover, many scholars tend to transform the problem of predicting changes of stock prices into the binary classification problem, which makes it difficult to earn steady abnormal returns. Therefore, it is extraordinary meaningful to study effective quantitative investment strategies. In this article, we propose the modified BP neural network combining AdaBoost algorithm (the modified BP_AdaBoost) and apply it into the quantitative stock selection. We carry out empirical studies about medium-term and long-term price changes in the A share market of our country, construct the factor pool and check the performances of the modified BP_AdaBoost.

1. Introduction
The factors affecting the stock market are various. In order to find appropriate methods to predict stock prices reasonably and effectively, scholars explored constructing a specific model, such as the Mean-Variance Model, Capital Asset Pricing Model, Value at Risk Model, Option Pricing Model and Arbitrage Pricing Model, and so on [1-3]. However, stock markets are the complex dynamic systems with the features of non-linearity, chaos and long-term memory. Therefore, it is difficult for us to use a specific model to make accurate predictions.

Taking those into account, scholars began to apply nonlinear theories to the fields of finance. With rapid developments of machine learning, the neural network model has been proved to be an effective tool for building nonlinear models. Scholars focused on the studies about the single neural network model. For example, Murat et al [4] and Oliveira et al [5] used the neural network model to predict the prices of the TKC securities and financial markets respectively. However the accuracy of prediction using the single neural network model is poor. In view of those, we can consider modifying the network structures or combining the model with other algorithms to improve the accuracy. For example, Khan et al [6] and Evans et al [7] combined genetic algorithms with neural networks and successfully found the optimal network topology. Zheng et al [8] proposed the RBF neural network with PSO to predict the stock prices and found that the convergences and the learning capacities of the model were good. Li et al [9] combined the AdaBoost algorithm with BP to forecast changes of the taxes, which improved the accuracy of prediction. We propose the quantitative stock selection model based on the modified BP_AdaBoost neural network and predict the rise and fall of prices in A share.
2. Methodology

2.1. The modified BP neural network

The BP neural network learns intrinsic characteristics of data through forward propagations of signals and reverse propagations of errors. During the forward propagation, factors enter into the input layer then are processed in hidden layers and transmitted to the output layer. If errors between actual values and expected output values in the output layer are too large or the times of learning aren’t reached, the reverse propagation of errors will begin. The reverse propagation of errors takes the output errors as bases to adjust weights of each hidden layers. Then again forward propagation starts. errors are calculated and the cycle repeats itself until the specified requirements are met. BP network structure is shown in Figure 1. In the $k$-th study training $\{(x_k, y_k), k = 1, \ldots, m\}$ where $x_k$, $y_k$ are input and output variables. We use $\hat{y}_k = (\hat{y}^1_k, \hat{y}^2_k, \ldots, \hat{y}^l_k)$, $\hat{y}^i_k = f(\beta_j - \theta_j)$, $j = 1, \ldots, l$ to denote the output variables of the neural network, where $l$ is the numbers of output neurons, and $\theta_j$ and $\beta_j$ are respectively the threshold and the input value received by the $j$-th neuron in the output layer. We let neuron activation function $f(\cdot)$ be the tangent hyperbolic function. The form is $f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$, and the diagram of curve is shown in Figure 2.

![Figure 1. The structure of BP neural network.](image1)

![Figure 2. The curve of the tangent hyperbolic function.](image2)

To unmodified BP neural networks, we need to minimize the following MSE (1), namely

$$E_k = \frac{1}{2} \sum_{j=1}^{l} (\hat{y}^j_k - y_k^j)^2.$$  \hspace{1cm} (1)

During learning processes of the single neural network, the over-fitting or under-fitting phenomenon often happen. To solve those problems, we modify the objective function (1) to improve its generalization by adding penalties. The MSE of the modified BP neural network becomes

$$E_k = \frac{1}{m} \sum_{k=1}^{m} E_k + \lambda J(\omega),$$  \hspace{1cm} (2)

which is called the structural risk formula, where $J(\omega)$ is the complexity of the model, $\lambda \geq 0$ is used to adjust empirical risks and model complexities. There are different forms of penalties which are used to express the complexity of the model, such as the $L_2$-norm and $L_1$-norm of the parametric vector $\omega$. They were called the Ridge regression and Lasso regression respectively in some articles [10]. Here we choose the ridge regression as the penalties, and then the formula (2) can be expressed by
\[ E_k = \lambda \frac{1}{m} \sum_{i=1}^{m} E_i + (1 - \lambda) \sum_i \omega_i^2, \quad \lambda \in (0, 1). \] 

2.2. The modified BP_AdaBoost neural network

Modified BP_AdaBoost is a kind of combined model. Figure 3 depicts the modified BP_AdaBoost system. Compared with combined models, the single modified BP may easily cause over-fitting or under-fitting problems, so here we call the single modified BP as the weak learner. By training weak learners, AdaBoost algorithm increases the weights of samples with poor training effects and weak learners with strong learning abilities, reduces the weights of samples with good training effects and weak learners with weak learning abilities, and then combines these weak learners linearly to improve generalization performances of the model. Weak learners with large differences can be improved obviously after using AdaBoost algorithm. If the single modified BP behaves well, the combined model will not has greater improvement in prediction accuracy. We randomly generate the modified BP predictors with low prediction accuracies to increase the differences between learners, and compare the prediction effects of different numbers of the weak learners. The algorithm of the modified BP_AdaBoost neural network is shown below in Table 1.

| Table 1. The algorithm of the modified BP_AdaBoost. |
|---------------------------------------------------|
| **Input:** The training data set \( T = \{(x_i, y_i),(x_2, y_2)\} \), the parameters of the weak learning \( m \) |
| **Output:** Predictor \( f(x) \) |
| 1: Initialize the weight distribution of training data \( D_1 = (\omega_{11}, \omega_{12}, \cdots, \omega_{1N}) \), \( \omega_i = 1/N, \ i = 1,2, \cdots, N \) |
| 2: \( t = 1 \) |
| 3: **While** \( t \leq m \) **do:** |
| 4: \( h \leftarrow \) generating the parameters of network hidden layers randomly |
| 5: Train \( ANN(h) \) as weak classifiers \( g_i(x) \) |
| 6: Calculate the maximum errors on training sets, \( E_i = \max (y_i - g_i(x_i)), \ i = 1,2, \cdots, N \) |
| 7: Calculate relative errors of each sample, \( e_i = (y_i - g_i(x_i)), \ i = 1,2, \cdots, N \) |
| 8: Calculate the prediction error rate, \( e_i = \sum_{i=1}^{N} Z_i e_i \) |
| 9: Calculate the coefficients of weak learners, \( \alpha_i = e_i / (1 - e_i) \) |
| 10: Update distributions of weights, \( \omega_{i+1} = \omega_i \alpha_i^{1+e_i} / Z_i \) and \( Z_i \) is the standardized factor, \( Z_i = \sum_{i=1}^{N} \omega_i \alpha_i^{1+e_i} \) |
| 11: Update the iterators, \( t = t + 1 \) |
| 12: **End.** Return learner \( g_i(x) \), coefficient \( \alpha_i \) |
| 13: Construct the final learner \( f(x) = \sum_{i=1}^{N} \alpha_i g_i(x) \) |

3. Empirical analysis

3.1. Data selection

The stock market is a dynamic system with strong noise, which can be easily influenced by external and internal factors. In order to improve the prediction accuracy of the modified BP_AdaBoost, we select stock data under common conditions and eliminate data with features skyrocketing and crashing. Figure 4 shows the movements of stock prices from January 4, 2010 to December 31, 2015. Referring to Kaufman (1998), we divided the original data into the training set, validation set and test set with the ratios of 70:15:15. In order to prevent time from disturbing the model predictions, the set is disorganized. The training set is the largest set, which is used to train the neural network and adjust weights. The validation set plays the auxiliary roles of building model by finding optimal parameters of the
neural network. The main function of test set is to check performances of models and evaluate generalization abilities of training networks. Here, we choose CSI 800 index as the object for our empirical research. Shanghai and Shenzhen stock markets involve most of prevailing investment stocks and have good representativeness.

Figure 3. The system of the modified BP_AdaBoost neural network.

3.2. Constructing the modified BP_AdaBoost factor pool
Various factors influence changes of stock prices. In this article, we construct the factor pool using those relatively mature and widely used factors. Just as the Table 2 shows, we use the Multi-factor stock selection strategy with the following seven factors.

Table 2. The factor pool.

| Types             | Factors                             | Types            | Factors         |
|-------------------|-------------------------------------|------------------|-----------------|
| Momentum factors  | Rates of return in Jan               | Valuation factors| PE              |
|                   | Rates of return in Feb               |                  | PCF             |
|                   | Rates of return in Mar               |                  | PS              |
|                   | Rates of return in Jun               |                  | CFFO            |
| Technological     | Ranges of prices in the first one months | Growth          | EPS             |
| factors           | Ranges of prices in the first two months | factors        | Growth rate of EPS |
|                   | Ranges of prices in the first three months | factors    | ROE              |
|                   | Ranges of prices in the first six months | factors      | Net Profits Change |
|                   | Volatility                          | Growth rate of EPS |
|                   | Amplitude                           | ROE              |
| Capital structure | Total value                         | Liquidity factors| Changes in turnover rate |
| factors           | Circulation market value            | Prices factor    | Ten day closing price |

Table 3. The confusion matrix.

| The confusion matrix | Prediction | Rise | Drop |
|----------------------|------------|------|------|
| Actuality            |            |      |      |
| Rise                 | TP         |      | FN   |
| Drop                 | FP         |      | TN   |

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \bar{y}_i)^2}$$  \hspace{1cm} (4)

$$F1 = \frac{2 \times P \times R}{P + R}$$  \hspace{1cm} (5)

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$  \hspace{1cm} (6)

$$P = \frac{TP}{TP + FP}$$  \hspace{1cm} (7)
We take the above factors as input values and the rise and fall of monthly average prices $y_i$ as output values, $y_i = p_i - p_{i+1}$, where $p_i$ is average monthly closing prices of stocks in the $i$-th month. Here we define if $y_i > 0$, the prices predicted by models will rise, otherwise, will decrease. Due to dimensions of variables are not uniform, we use "maximum-minimum standardization" method to transform data into values of $[0,1]$. After testing, we let the learning rate be 0.005, the initial weights and thresholds be generated as $U[0,1]$ and times of cycles be 2000. We construct the confusion matrix (Table 3) and the indexes for checking model performances are shown above.

3.3. Empirical results
The numbers of weak learners will affect performances of prediction. From Table 4 and Figure 5, we can see the classification accuracies using a small amount of weak learners are poor. Even numbers of learners increase, Accuracy and $p$ are around 78% and 77% without obvious changes. If we set numbers of weak learners appropriately, RMSE will be smaller. Synthesize behaviors of the fittings on the validation data and test data, we find when the number of learners is 20, the accuracy and stability of the modified BP_AdaBoost neural network are excellent, so we let the number of weak learners be 20.

Table 4. Prediction results of the modified BP_AdaBoost.

| Numbers | The validation set | The test set |
|---------|-------------------|--------------|
|         | Accuracy  | F1  | P   | RMSE  | Accuracy  | F1  | P   | RMSE  |
| 5       | 0.7850    | 0.7729 | 0.7825 | 1.0170 | 0.7758    | 0.7655 | 0.7756 | 0.9189 |
| 10      | 0.7817    | 0.7706 | 0.7760 | 1.0187 | 0.7783    | 0.7691 | 0.7758 | 1.0036 |
| 15      | 0.7808    | 0.7727 | 0.7727 | 1.0532 | 0.7767    | 0.7674 | 0.7741 | 1.1046 |
| 20      | 0.7833    | 0.7723 | 0.7778 | 1.0031 | 0.7767    | 0.7674 | 0.7741 | 1.0533 |
| 30      | 0.7833    | 0.7723 | 0.7778 | 1.0568 | 0.7775    | 0.7676 | 0.7764 | 1.0437 |
| 50      | 0.7825    | 0.7713 | 0.7774 | 1.0084 | 0.7783    | 0.7687 | 0.7758 | 1.0140 |

Figure 5. Prediction effect of the modified BP_AdaBoost.

To analyze prediction efficiencies of the modified BP_AdaBoost further, we compare the following four indexes, Accuracy, $p$, RMSE and F1 between BP and the modified BP_AdaBoost. As shown in Table 5, evaluation indexes of the modified BP_AdaBoost are better than those of BP. In detail, in the validation set, Accuracy of the modified BP_AdaBoost is higher than that of BP. In the test set, Accur-
acies of the two models are close. Our strategy is that if those prices predicted by models are up, we tend to buy stocks and believe that prices will fall, otherwise, we will not buy them and believe prices to be down. That is, when prices predicted by the model go up but actual values are down, which will bring serious adverse impacts to earnings. Hence, we attach great importance to the values of \( P \). In the verification set and test set, values of \( P \) of the modified BP_AdaBoost are higher than those of BP, that is to say, the modified BP_AdaBoost has fewer false positive conditions in prediction results and stock selection abilities of the model are better. Similarly, RMSEs of the modified BP_AdaBoost are smaller than those of BP, and the fitting effects are obviously improved. In conclusion, our proposed modified BP_AdaBoost neural network behaves better.

Table 5. The prediction results of different network models.

| Type               | The validation set | The test set |
|--------------------|--------------------|--------------|
|                    | Accuracy | P   | RMSE | F1    | Accuracy | P   | RMSE | F1    |
| Modified BP_AdaBoost | 0.7850   | 0.7825 | 1.0170 | 0.7729 | 0.7758   | 0.9189 | 0.7655 |
| BP                 | 0.7783   | 0.7409 | 2.3158 | 0.7621 | 0.7792   | 2.1145 | 0.7657 |

4. Conclusion
In this article, we propose the modified BP_AdaBoost neural network, using the \( L_2 \) norm as penalties to modify BP and combining AdaBoost algorithm. We apply the modified BP_AdaBoost to empirical studies about the A shares. Construct the factor pool with mainstream indexes in the financial market. The results show our proposed modified BP_AdaBoost performs well in quantitative stock selection.

Acknowledgments
This article is supported by Natural Science Foundation of Zhejiang Province (No. LY17A010002); National Social Science Foundation of China (No. 17BTJ027); first-class disciplines of Zhejiang Province, characteristic disciplines of Zhejiang Province and Statistics of Zhejiang Gongshang university.

References
[1] Gady J, David J F and Aron A G 2000 J. The capital asset pricing model and the liquidity effect: A theoretical approach 3 69-81
[2] Zhang Z Y, Kiyotaka S and Michael M 2004 J. Asian monetary integration: a structural VAR approach 64 447-458
[3] He Z L and Xu Q 2015 J. Portfolio Choice under the Mean-Variance Model with Parameter Uncertainty 32 498-503
[4] Murat A and Ozbayoglu 2008 J. Neural based technical analysis in stock market forecasting Intelligent Engineering Systems through Artificial Neural Networks 45 261-265
[5] Oliveira F A D, Nobre C N and Zarate L E 2013 J. Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index-Case study of PET-R4, Petrobras, Brazil 40 7596-7606
[6] Khan A U, Bandopadhyaya T K and Sharma S 2008 Conf. Comparisons of Stock Rates Prediction Accuracy Using Different Technical Indicators with Back propagation Neural Network and Genetic Algorithm Based Back propagation Neural Network (Nagpur: IEEE) pp 575-80
[7] Evans C, Pappas K and Xhafa F 2013 J. Utilizing artificial neural networks and genetic algorithms to build an algo-trading model for intra-day foreign exchange speculation 58 1249-66
[8] Zheng R Y and Wu Y H 2011 J. A Research on Neural Network in Stock Prices Prediction 28 393-396
[9] Li X and Zhu Q Y 2012 J. Tax Prediction Based on AdaBoost and BP Neural Network 32 3558-60
[10] Vidyasagar M 1998 J. An introduction to some statistical aspects of PAC learning theory 34 115-124