

Research into Financial Position of Listed Companies following Classification via Extreme Learning Machine Based upon DE Optimization

Yu Fu¹, Jiong Mu^{1,a} and Xu Liang Duan¹

¹ College of Information Engineering and Technology Sichuan Agricultural University, Ya'an, China

Abstract. By means of the model of extreme learning machine based upon DE optimization, this article particularly centers on the optimization thinking of such a model as well as its application effect in the field of listed company's financial position classification. It proves that the improved extreme learning machine algorithm based upon DE optimization eclipses the traditional extreme learning machine algorithm following comparison. Meanwhile, this article also intends to introduce certain research thinking concerning extreme learning machine into the economics classification area so as to fulfill the purpose of computerizing the speedy but effective evaluation of massive financial statements of listed companies pertain to different classes

1 Introduction

For investors, the financial statements of listed companies constitute an important source of information from which they can understand the current financial status of those listed companies and reasonably predict their development tendency. Unfortunately, due to the financial statements containing lots of hidden information and parameters of various indicators, it is hard to analyze them and therefore, classifying listed company's financial position via directly observing their financial statements becomes more and more tough.

The purpose of this article is to put forth a model of extreme learning machine based upon DE optimization that is capable of modeling the financial status of a listed company after learning and analyzing its financial statement samples in a given quantity and then classifying the enormous financial data of other listed companies imported later on so as to save the time spent on classification and optimize the classification effect. What's special about this article is to rest the extreme learning machine algorithm on DE optimization in the first place and then apply it to classifying listed company's financial position. By the way, there are several classification algorithms having been precedentedly proposed, for example, the Five-Class Determination Method suggested by Lv Changjiang^[2] and others, which suggests to group the financial situation of listed companies into the following 5 classes: financial idleness, financial adequacy, financial balance, financial difficulty and financial bankruptcy, the Support Vector Machine (SVM) Classification Algorithm recommended by Song Jiao^[3] that establishes a SVM-based financial crisis pre-warning model for listed companies to effectively prevent financial crisis, and the BP Neural Network Pre-warning Model brought forth by Yang Shu'e^[4] and others, which similarly aims to help listed companies evade financial crisis. Nevertheless, articles focusing upon utilizing intelligent algorithm to quickly classify the financial statements of numerous listed companies are rather insufficient.

2 DE(Differential Evolution)optimization

^a Corresponding author: scmjmj@163.com

As an emerging evolutionary computation technology initiated by Storn^{[5]-[7]} and others in 1995, DE, short for Differential Evolution, is, just like many other evolutionary algorithms, a random model simulating biological evolution through which individuals more adaptive to the surroundings finally survive via repeated iteration. Thanks to its powerful global convergence ability, robustness and independence to the characteristic information of issues, DE can be employed to solve the optimization question in a tough environment that is impossible to be cracked by means of common digital planning method. The following is the algorithm thinking of DE:

First of all, randomly initiate the population in the issue's feasible region $X^0 = [x_1^0, x_2^0, \dots, x_{Np}^0]$, Np is population scale. This individual

$x_i^0 = [x_{i,1}^0, x_{i,2}^0, \dots, x_{i,D}^0]$ is used to solve characteristic

issues and D is dimension of the optimization issue. The basic idea of the algorithm is as follows: Generate a new population through mutating and intersecting the current population and subsequently, choose both populations one-on-one via the fundamental greedy thinking to ultimately give birth to a new population. To be more exact, mutate each individual x_i^t at the time of t by Formula (1) to obtain the corresponding mutated individual x_i^{t+1} , i.e.:

$$v_i^{t+1} = x_{r_1}^t + k(x_{r_2}^t - x_{r_3}^t) \quad (1)$$

Wherein $r_1, r_2, r_3 \in \{1, 2, \dots, N_p\}$ are distinguished from each other as well as i ; $x_{r_1}^t$ is father basic vector, $(x_{r_2}^t - x_{r_3}^t)$ is father differential vector and K is scaling factor. Intersect x_i^t and the mutated individual v_i^{t+1} derived from Formula (1) by Formula (2) to generate the experimental individual u_i^{t+1} , i.e.:

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}, & \text{If } (rand(j) \leq CR) \text{ or } j = mbr(i); \\ x_{i,j}^t, & \text{Otherwise.} \end{cases} \quad (2)$$

Wherein $rand(j)$ is evenly distributed in $[0, 1]$ at random, CR is crossover probability within the range of $[0, 1]$ and $mbr(i)$ is random quantity in $\{1, 2, \dots, D\}$. Compare the target function of the experimental individuals u_i^{t+1} and x_i^t via Formula (3). Regarding the minimization issue, accept the individuals with low target function value as the individual of new population x_i^{t+1} , i.e.:

$$x_i^{t+1} = \begin{cases} u_i^{t+1}, & \text{If } f(u_i^{t+1}) < f(x_i^t); \\ x_i^t, & \text{Otherwise.} \end{cases} \quad (3)$$

Wherein f is target function. The algorithm mentioned before is the DE algorithm model discussed in this article^[8]. The general thinking of this article is to carry out data optimization in the DE algorithm model and then import the optimized data into the extreme learning machine model for the purpose of classification.

3 ELM (Extreme Learning Machine) classification

Introduced in 2004 by Huang Guangbin, an associate professor of Nanyang Technological University, the extreme learning machine mentioned herein is a Single-Hidden Layer Feedforward Neural Networks (SLFNs) learning algorithm^{[9]-[12]} that is characterized by fast learning speed and excellent generalization^{[13],[16] and [17]}. As for its algorithm thinking, please refer to as follows:

For a single-hidden layer neural network, assume there are N random samples (X_i, t_i) , wherein:

$$\begin{aligned} X_i &= [x_{i_1}, x_{i_2}, \dots, x_{i_n}]^T \in R^n, \\ t_i &= [t_{i_1}, t_{i_2}, \dots, t_{i_m}]^T \in R^m. \end{aligned} \quad (4)$$

A single-hidden neural network containing \tilde{N} hidden layer nodes can be expressed as follows:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(W_i \cdot X_j + b_i) = o_j, \quad j = 1, \dots, N \quad (5)$$

In Formula (5), $g(x)$ is activation function, $W_i = [w_{i,1}, w_{i,2}, \dots, w_{i,n}]^T$ is input weight, β_i is output weight, and b_i is offset of the i hidden layer unit. $W_i \cdot X_j$ represents the inner product of W_i and X_j .

The purpose the single-hidden neural network learning intends to serve is to minimize the tolerance of output, which can be expressed in Formula (6):

$$\sum_{j=1}^{\tilde{N}} \|o_j - t_j\| = 0 \quad (6)$$

Namely, it is β_i , W_i and b_i that make Formula (7) established:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(W_i \cdot X_j + b_i) = t_j, \quad j = 1, \dots, N \quad (7)$$

$$\text{Expressed via matrix: } H\beta = T \quad (8)$$

In Formula (8), H is output of hidden layer node, β is output weight and T is expected output, i.e. there are three Formulas (9), (10) and (11) given below:

$$H(W_1, \dots, W_{\tilde{N}}, b_1, \dots, b_{\tilde{N}}, X_1, \dots, X_N) = \begin{bmatrix} g(W_1 \cdot X_1 + b_1) & \dots & g(W_{\tilde{N}} \cdot X_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(W_1 \cdot X_N + b_1) & \dots & g(W_{\tilde{N}} \cdot X_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \quad (9)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad (10), \quad T = \begin{bmatrix} T_1^T \\ \vdots \\ T_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad (11)$$

In order to train the single-hidden layer neural network, we hope to obtain \hat{W}_i , \hat{b}_i and $\hat{\beta}$ to make Formula (12) established:

$$\|H(\hat{W}_i, \hat{b}_i)\hat{\beta} - T\| = \min_{W, b, \beta} \|H(W_i, b_i)\beta - T\| \quad (12)$$

In Formula (12), $i = 1, \dots, \tilde{N}$ means that the loss minimization function (13) is established.

$$E = \sum_{j=1}^N \left(\sum_{i=1}^{\tilde{N}} \beta_i g(W_i \cdot X_j + b_i) - t_j \right)^2 \quad (13)$$

In the ELM algorithm, once the input weight W_i and the hidden layer offset b_i have been determined randomly, the output matrix in the hidden layer H will be solely decided. Meanwhile, training a single-hidden neural network can be converted into the process of solving a linear system: $H\beta = T$ and the output weight β can be confirmed as $\hat{\beta} = H^T T$, wherein H^T is the Moore-Penrose generalized inverse of the matrix H and the solution $\hat{\beta}$ proves to be mere and minimum.

4 Data collection

In order to better explain the model mentioned in this article and back up the actual application effect of such a model, we have extracted the financial statements of 200 listed companies from the annual report of Chinese listed companies as samples ^[14]. Among those companies, 87 and 113 companies are respectively listed in Shenzhen Stock Exchange and Shanghai Stock Exchange. Total 49 financial indicators ^[15] such as Earnings per Share, Net Profit and Debt to Assets Ratio etc. included in the financial statements are adopted as characteristic information to comprehensively assess the financial position of those companies and then classify them into the following five different levels: Excellent, Good, Normal, Unsatisfactory and Terrible. For the foregoing financial indicators, please refer to the table below:

Table 1. Financial Indicators Chosen as Characteristic Values

49 Indicators	
Primary Earnings per Share (EPS)	Current Liabilities /Liabilities in Total
Book value per Share	Asset Fixation Ratio
Total Operating Revenue per Share	Liquidity Ratio
Operating Revenue per Share	Quick Ratio
Capital Reserve per Share	Conservative Quick Ratio
Surplus Reserve per Share	Cash Ratio
Undistributed Profits per Share	Equity Belonging to Parent Company's Shareholders /Total Liabilities
Retained Earnings per Share	Tangible Assets/ Total Liabilities
Net Cash Flow per Share	Debt to Tangible Assets Ratio
Earnings before Interest and Tax per Share	Primary Earnings per Share (Year-on-Year Growth Rate)
Return on Total Assets (ROA) Ratio	Net Cash Flow Generated by Operating Activities per Share (Year-on-Year Growth Rate)
Net Return on Total Assets (ROA) Ratio	Total Operating Revenue (Year-on-Year Growth Rate)
Annualized Return on Total Assets (ROA) Ratio	Operating Revenue (Year-on-Year Growth Rate)
Annualized Net Return on Total Assets (ROA) Ratio	Operating Profit (Year-on-Year Growth Rate) 1
Net Profit Margin on Sales	Operating Profit (Year-on-Year Growth Rate) 2
Gross Profit Margin on Sales	Net Profit (Year-on-Year Growth Rate)
Ratio of Sales to Cost	Net Profit Belonging to

	Parent Company's Shareholders (Year-on-Year Growth Rate)
Expense Ratio in Sales Period	Net Profit Belonging to Parent Company's Shareholders Minus Extraordinary Gain or Loss (Year-on-Year Growth Rate)
Main Business Ratio	Net Cash Flow Generated by Operating Activities (Year-on-Year Growth Rate)
Debt to Assets Ratio	Monetary Fund Growth Rate
Long-term Assets Suitability Ratio	Fixed Assets Investment Expansion Ratio
Current Assets/Total Assets	Net Assets (Year-on-Year Growth Rate)
Current Debt-to-Equity Ratio	Total Liabilities (Year-on-Year Growth Rate)
Non-current Assets/Total Assets	Total Assets (Year-on-Year Growth Rate)
Tangible Assets/ Total Assets	

5 Comparison between Extreme Learning Machine Classification Algorithm Based-upon DE Optimization and Traditional Extreme Learning Machine Classification Algorithm

We divided those 200 listed companies into the following 2 parts, one is made up of 140 companies that have been respectively imported into the traditional extreme learning machine and the extreme learning machine based upon DE optimization in the Matlab environment, and the other is the remaining 60 companies whose financial statements are utilized as test samples to observe the classification situation. Regarding the first part, the above-mentioned 49 financial indicators are considered algorithm characteristic information and the corresponding extreme learning machine models have been established and trained. Finally, compare the classification conditions of both algorithms and the related results are shown below:

5.1. Results of both classification algorithms

Through MATLAB simulation, we can observe the results of both classification algorithms as shown below:

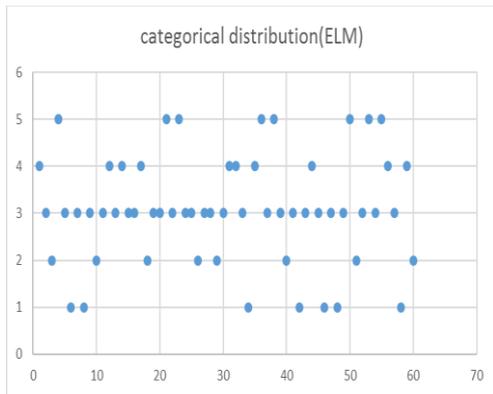


Figure 1. Classification results of the ELM algorithm

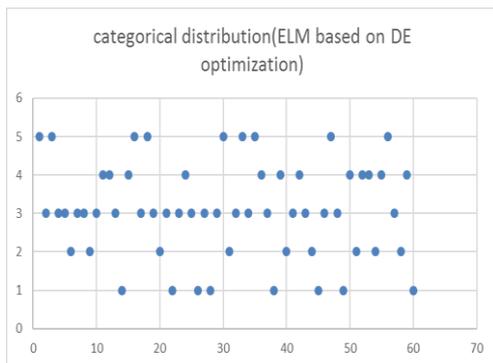


Figure 2. Classification results of the DE optimization-related ELM algorithm

The vertical coordinates shown in Figures 1 and 2 suggest that 1 denotes terrible financial situation, 2 unsatisfactory situation, 3 normal financial situation, 4 good financial situation and 5 excellent financial situation.

The classification results delineated in Figure 1 are generated by the traditional extreme learning machine classification algorithm, to be exact, the financial situation of 7 listed companies is regarded as terrible (Class 1), 8 unsatisfactory (Class 2), 27 normal (Class 3) and 10 good (Class 4) and 8 excellent (class 5). The general classification error in connection with the algorithm mentioned before is 12.7% when comparing with the actual financial situation of those listed companies, namely, 7 out of 60 listed companies are incorrectly classified, which renders the correct recognition rate of such an algorithm reach 88% more or less.

The classification results delineated in Figure 2 are generated by the DE optimization-related extreme learning machine classification algorithm, to be exact, the financial situation of 8 listed companies is regarded as terrible (Class 1), 9 unsatisfactory (Class 2), 22 normal (Class 3) and 12 good (Class 4) and 9 excellent (class 5). There are 4 out of 60 listed companies incorrectly classified, thus making the incorrect recognition rate of such an algorithm reach 6.7% that is rather less than the traditional extreme learning machine classification algorithm.

5.2 The training performance of both algorithm models is presented below

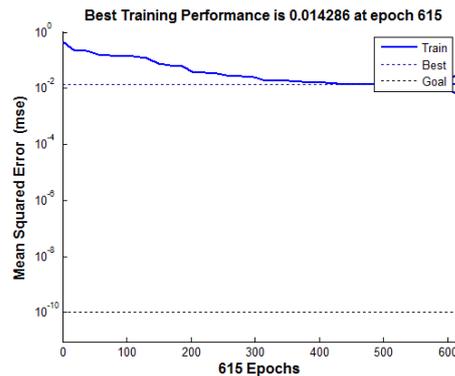


Figure 3. Training performance of the ELM algorithm

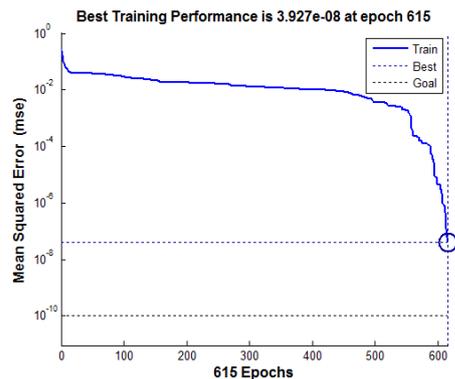


Figure 4. Training performance of the DE optimization-related ELM algorithm

In order to facilitate comparison, we have iterated both algorithms 615 times to see their error performance. Judging by Figure 3, for the traditional extreme learning machine, the Mean Square Error infinitely approaches 10^{-2} following 615 times of iteration. On the other hand, as revealed in Figure 4, for the DE optimization-related extreme learning machine, the Mean Square Error is able to reach 10^{-8} also following 615 times of iteration, which is slightly different from the target error we anticipated, i.e. 10^{-10} . To judge from what's described before, the DE optimization-related extreme learning machine algorithm is capable of effectively reducing the tolerance of data classification when comparing with the traditional extreme learning machine algorithm, accordingly elevating the accuracy in connection with data classification to an ideal status.

5.3 Comparison between sample data and ideal output results in terms of fitting degree

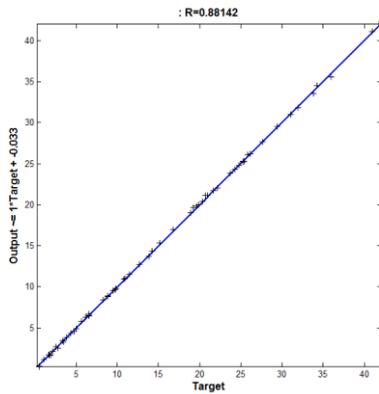


Figure 5. Comparison between sample output by the ELM algorithm and ideal output in terms of fitting degree

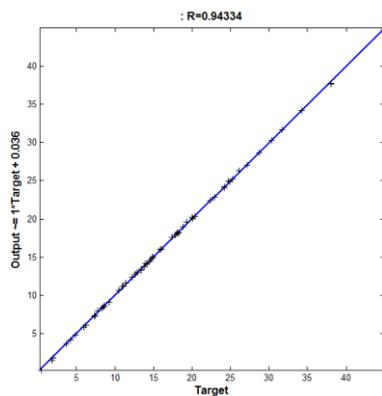


Figure 6. Comparison between sample output by the DE optimization-related ELM algorithm and ideal output in terms of fitting degree

In Figures 5 and 6, a corresponding idealized output model is built by putting the financial data of those 140 listed companies into the 2 different algorithm models so as to generate the Slope $k=1$ (vertical coordinate/horizontal coordinate=1), an ideal output straight line, i.e. this line entirely complies with the ideal output. In the left upper part and the right lower part of such a straight line, there are discrete intervals, namely, the further away the mark + in Figures 5 and 6 is from such a straight line, the bigger the difference between the output results of the experimental data and the ideal results becomes. On contrary, the closer the mark + approaches the ideal output line $K=1$, the smaller difference between the output results of the experimental data and the ideal results becomes, i.e. the higher the fitting degree turns. The mark + in Figures 5 and 6 represents that the actual classification results of financial data in connection with those 60 listed companies are in positive correlation with the fitting degree of the output samples, which means the higher the sample's output fitting degree is, the more accurate the classification results become.

For the traditional extreme learning machine model in Figure 5, its output fitting degree ($R=0.88142$) doesn't satisfactorily approach the ideal output results as we hoped since the fitting degree tolerance is as follows:

$\Delta=1-0.88142=0.11858$. For the DE optimization-related extreme learning machine model in Figure 6, the output fitting degree ($R=0.94334$) is obviously close to the ideal output results as the fitting degree tolerance is below:

$$\Delta=1-0.94334=0.05666.$$

Additionally, it is predictable that as the training sample quantity and the time of iteration continuously increase, the DE optimization-related extreme learning machine algorithm is capable of making the experimental sample's fitting degree tolerance further close to ideal output, which can be seen from the training performance under the condition of same iteration times.

6 Conclusion

In this article we have proposed a DE optimization-based extreme learning machine algorithm that can be deployed to classify the financial position of different listed companies. After classifying and analyzing the financial data of those 200 listed companies and comparing with the traditional extreme learning machine algorithm, the algorithm described in this article, i.e. the DE optimization-based extreme learning machine algorithm, proves to be improved in terms of accuracy and superiority. At the same time, this article also attempted to introduce a new analysis thinking into the economic field, in other words, freeing humans from tedious and repetitive labor via a highly-efficient smart algorithm catering for a great deal of data and applying the multi-characteristic classification method to solving economic analysis issues so as to considerably cut down the time spent on financial information classification. However, this article is restricted by inadequate training and test samples. The test sample recognition rate would be higher if the training samples increased in amount.

References

1. Altman , E. , 1968 , Financial Ratios , Discriminate Analysis and the Prediction of Corporate Bankruptcy , Journal of Finance 23(Sept .)
2. Changjiang Lv, Yan Zhao, Classification of financial status of listed companies. J. Accounting Research, 11(2004).
3. Song Jiao, Corporate Financial Distress on support vector machine .J, Harbin University of Commerce (Natural Science), 23(2007)
4. .Shue Yang, Li Huang, financial early warning model of listed companies BP neural network .J, Based on System Engineering Theory and Practice, 25(2005)
5. Storn R, Price K,.Differential evolution ---- A simple and efficient adaptive scheme for global optimization over continuous spaces,R, Berkeley: University of California,(2006).
6. Storn R, Price K. Minimizing the real functions of the ICEC'96 contest by differential evolution.C, Proc of IEEE Int Conf on Evolutionary Computation. Nagoya,(1996).

7. Bo Liu, Ling Wang, Yihui Jin, Research differential evolution algorithm.J, Control and Decision, 22(2007)
8. Shengshuang Chen, based on XML documents ultimate learning machine classification .J, Computer Engineering, 37(2011).
9. G.-B. Huang, et al., Universal Approximation Using Incremental Networks with Random Hidden Computational Nodes, IEEE Transactions on Neural Networks, 4(2006).
10. G.-B. Huang, et al., Extreme Learning Machine: Theory and Applications, Neurocomputing, (2006).
11. N.-Y. Liang, et al., A Fast and Accurate On-line Sequential Learning Algorithm for Feedforward Networks, IEEE Transactions on Neural Networks, 6,17(2006).
12. G.-B. Huang, et al., Can Threshold Networks Be Trained Directly?, IEEE Transactions on Circuits and Systems II, 3,53(2006).
13. K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, *Neural Netw.*, 5,2(1989).
14. H. jun Rong, Y.-S. Ong, A.-W. Tan, and Z. Zhu, A fast pruned-extreme learning machine for classification problem, *Neurocomputing*, 1,72(2008).
15. G. Bontempi, M. Birattari, and H. Bersini, Recursive lazy learning for modeling and control, in *Proc. Eur. Conf. Mach. Learn.*, (1998).
16. Shizhong Liao, Chang Feng. Meta-ELM: ELM with ELM hidden nodes, J, Neurocomputing, (Mar.27) (2014).
17. Extreme Learning Machines (Trends & Controversies).J, IEEE intelligent systems, 28(2013).

Acknowledgment

This paper was funded by Sichuan Provincial Education Department ,project name: Building Agricultural Internet of Things hierarchical storage model(15ZB0017).