Neurocomputing 128 (2014) 296–302

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Bankruptcy prediction using Extreme Learning Machine and financial expertise

Qi Yu [a,*], Yoan Miche [a], Eric Séverin [b], Amaury Lendasse [a,c,d]

[a] Department of Information and Computer Science, Aalto University, FI-00076 Espoo, Finland
[b] LEM, Université Lille 1, 59043 Lille Cedex, France
[c] IKERBASQUE, Basque Foundation for Science, 48011 Bilbao, Spain
[d] Computational Intelligence Group, Computer Science Faculty, University of the Basque Country, Paseo Manuel Lardizabal 1, Donostia-San Sebastián, Spain

ARTICLE INFO

Article history:
Received 31 August 2012
Received in revised form
25 December 2012
Accepted 30 January 2013
Available online 24 October 2013

Keywords:
Extreme Learning Machine
Leave-One-Out
Incremental Learning
Bankruptcy Prediction

ABSTRACT

Bankruptcy prediction has been widely studied as a binary classification problem using financial ratios methodologies. In this paper, Leave-One-Out-Incremental Extreme Learning Machine (LOO-IELM) is explored for this task. LOO-IELM operates in an incremental way to avoid inefficient and unnecessary calculations and stops automatically with the neurons of which the number is unknown. Moreover, Combo method and further Ensemble model are investigated based on different LOO-IELM models and the specific financial indicators. These indicators are chosen using different strategies according to the financial expertise. The entire process has shown its good performance with a very fast speed, and also helps to interpret the model and the special ratios.

© 2013 Published by Elsevier B.V.

1. Introduction

The business failure has been widely studied, trying to identify the various determinants that can affect the (Non) existence of firms. Especially due to the recent changes in the world economy and as more firms, large and small, seem to fail now more than ever. The prediction of the bankruptcy is then of increasing importance.

In most of the studies, bankruptcy prediction is treated as a binary classification problem. The target (output) variable of the models is commonly a dichotomous variable where “firm filed for bankruptcy” is set to 1 and “firm remains solvent” is set to 0. The reference (input) variables are often financial ratios drawn from financial statements and include measures of profitability, liquidity, and leverage. The pioneer study using univariate statistic of financial ratios originated from Beaver [1] and Altmans work [2]. Using multivariate discriminate analysis to assess predictive power of ratio analysis, financial ratios methodologies are becoming indispensable tools for modeling, analysis and prediction. The other main stream is employing Artificial Intelligence (AI) methods, which have been applied to bankruptcy prediction problem from 1990s, including decision tree [3,4], fuzzy set theory, case-based reasoning [5], genetic algorithm [6], support vector machine, several kinds of neural networks such as BPNN (back propagation trained neural network), PNN (probabilistic neural networks) [7], SOM (self-organizing map). However, all those mentioned methods including existing financial models suffer from the problems of strict hypothesis, poor generalization ability, low prediction accuracy and low learning rate, slow computational time, etc. To overcome these, a newly popular method Extreme Learning Machine (ELM) is considered in this paper.

Extreme Learning Machine (ELM), which is a simple and efficient learning algorithm for single-hidden layer feedforward neural networks (SLFNs), has been recently proposed in [8]. ELM has shown good generalization performances for many real applications with an extremely fast learning speed [9–15]. However, like other similar approaches based on feedforward neural networks, some issues with the practical applications of the ELM still arise, most importantly, how to obtain the most appropriate architecture of the network. In other words, how to select or search for the optimal number of hidden neurons remains a difficult problem.

Many methods have been exploited recently trying to choose the most suitable network structure of ELM and to further reduce the number of neurons without affecting the generalization performance. Pruning methods are one type of algorithms to address this problem. For example, Rong et al. in [16] proposed a pruned ELM (P-ELM) for classification, and Miche et al. in [17,14] presented a method called optimally pruned ELM (OP-ELM). But pruning methods in general are rather inefficient since most of the

* Corresponding author. Tel.: +358 442 519 088.
E-mail address: qi.yu@aalto.fi (Q. Yu).

0925-2312/$ - see front matter © 2013 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.neucom.2013.01.063

time they are dealing with a network structure larger than necessary. On the other hand, some researchers manage to solve the problems via incremental learning. Like the Incremental Extreme Learning Machine (I-ELM) [18] which adds randomly generated hidden nodes one-by-one to the hidden layer until achieving an expected training accuracy or reaching the maximum number of hidden nodes. There are also some modifications made to I-ELM, like shown in [19–21]. However, these methods need to set the expected training error or maximum number of neurons in advance.

In this paper, a method called Leave-One-Out-Incremental Extreme Learning Machine (LOO-IELM) is proposed. It is operated in an incremental way but stops automatically based on the stop criteria. Besides, Combo method and Ensemble model are explored based on LOO-IELM, as well as some preknowledge of financial expertise in order to improve the bankruptcy prediction accuracy. In the experiments, all methods are tested on both two sets (9 *priori* selected Variables from automatic black-box variable selection [22] and 12 *priori* selected Variables from financial expertise [23]).

2. Extreme Learning Machine

The Extreme Learning Machine algorithm is proposed by Huang et al. in [8] as an original way of building a single Hidden Layer Feedforward Neural Network (SLFN). The essence of ELM is that the hidden layer needs not be iteratively tuned [24,8], and moreover, the training error $\|\mathbf{H}\beta-\mathbf{y}\|$ and the norm of the weights $\|\beta\|$ are minimized.

Given a set of N observations $(x_i, y_i), i \le N$. with $x_i \in \mathbf{R}^p$ and $y_i \in \mathbf{R}$. A SLFN with m hidden neurons in the hidden layer can be expressed by the following sum:

$$\sum_{i=1}^{m} \beta_i f(\omega_i x_j + b_i), \quad 1 \le j \le N \tag{1}$$

where β_i are the output weights, f be an activation function, ω_i the input weights and b_i the biases. Suppose the model perfectly describes the data, the relation can be written in the matrix form as $\mathbf{H}\boldsymbol{\beta} = \mathbf{y}$, with

$$\mathbf{H} = \begin{pmatrix} f(\omega_1 x_1 + b_1) & \dots & f(\omega_m x_1 + b_m) \\ \vdots & \ddots & \vdots \\ f(\omega_1 x_n + b_1) & \dots & f(\omega_m x_n + b_m) \end{pmatrix} \tag{2}$$

$\boldsymbol{\beta} = (\beta_1, \dots, \beta_m)^T$ and $\mathbf{y} = (y_1, \dots, y_n)^T$. The ELM approach is thus to initialize randomly the ω_i and b_i and compute the output weights $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{y}$ by a Moore–Penrose pseudo-inverse [25].

The significant advantages of ELM are its extremely fast learning speed, and its good generalization performance while being a simple method [8]. There has been recent advances based on the ELM algorithm to improve its robustness (OP-ELM [17], TROP-ELM [14], CS-ELM [26]), or make it a batch algorithm, improving at each iteration (EM-ELM [27], EEM-ELM [28]).

Along with the increase of number of hidden nodes in ELM, the error usually decreases. However, it also brings the corresponding difficulty: the complexity of the model and how to optimize between both.

3. Incremental Extreme Learning Machine with Leave-One-Out (LOO-IELM)

In this section, a method called LOO-IELM is presented as well as the details of the implementation steps. In general, the method can be operated as the following:

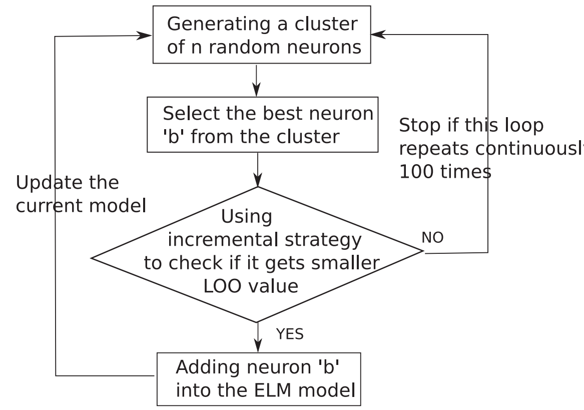

Fig. 1. The framework of the LOO-IELM method.

Fig. 1 illustrates the main procedures of LOO-IELM, and how they interact. The neurons of the ELM are selected and added incrementally till no smaller LOO value can be found. Next paragraphs concentrate on more details of this method.

Given a training set $(x_i, y_i) | x_i \in \mathbf{R}^d, y_i \in \mathbf{R}, i = 1, 2, \dots, d$, activation function $f(x)$. $\mathbf{H}$ represents the output matrix of the hidden layer. Each trial cluster contains n neurons.

3.1. Incremental strategy

Initialization step: Let the number of hidden neurons be zero at the very beginning, then the neurons could be chosen progressively later on by LOO-IELM.

Learning step:

- Randomly generate a cluster of n neurons. n is optional that can be configured according to the different computer power or different data sets. It saves computational time to test neurons cluster by cluster, than one by one. In this paper, n is chosen to be 20.
- Construct ELM using the combination of each of the n neurons and the existing selected neurons. That means ELM models are build 20 times in this step. (For the first round, it means to construct ELM with each neuron separately.) Test the LOO value for each of these ELMs, find the neuron "b" that gives the smallest LOO.
- Check whether the LOO value with neuron "b" is smaller than previous one. If so, continue to next step; otherwise, stop current trial and repeat the learning step.

Stop criterion: One advantage of LOO-IELM is that no parameter needs to be set beforehand, the number of neurons is chosen automatically according to the algorithm. Therefore, when to stop finding new neurons becomes an issue for this method. In this paper, the default setting is 100 extra clusters. As we mentioned that the neurons are tested cluster by cluster, instead of one by one in other incremental learning algorithm. Therefore, this means LOO-IELM stops training if LOO value does not decrease for continuous 2000 new neurons (here each cluster contains $n=20$ neurons).

3.2. Leave-One-Out (LOO)

The decision over the actual selected neurons for the model is taken using a Leave-One-Out method. One problem with the LOO error is that it can get very time consuming if the data set tends to have a high number of samples. Fortunately, the PRESS (or PREdiction Sum of Squares) statistics provide a direct and exact formula for the calculation of the LOO error for linear models. See

Download English Version:

https://daneshyari.com/en/article/408156

Download Persian Version:

https://daneshyari.com/article/408156

Daneshyari.com