Applied Soft Computing 24 (2014) 977–984

Contents lists available at ScienceDirect

Applied Soft Computing

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

A comparative study of classifier ensembles for bankruptcy prediction

Chih-Fong Tsai[a,1], Yu-Feng Hsu[b,2], David C. Yen[c,*]

[a] Department of Information Management, National Central University, Jhongli, Taiwan, ROC
[b] Department of Information Management, National Sun Yat-Sen University, Kaohsiung, Taiwan, ROC
[c] School of Economics and Business, 226 Netzer Administration Building, SUNY College at Oneonta, Oneonta, NY 13820, United States

ARTICLE INFO

Article history:
Received 20 March 2013
Received in revised form 8 April 2014
Accepted 22 August 2014
Available online 6 September 2014

Keywords:
Bankruptcy prediction
Credit scoring
Classifier ensembles
Data mining
Machine learning

ABSTRACT

The aim of bankruptcy prediction in the areas of data mining and machine learning is to develop an effective model which can provide the higher prediction accuracy. In the prior literature, various classification techniques have been developed and studied, in/with which classifier ensembles by combining multiple classifiers approach have shown their outperformance over many single classifiers. However, in terms of constructing classifier ensembles, there are three critical issues which can affect their performance. The first one is the classification technique actually used/adopted, and the other two are the combination method to combine multiple classifiers and the number of classifiers to be combined, respectively. Since there are limited, relevant studies examining these aforementioned disuses, this paper conducts a comprehensive study of comparing classifier ensembles by three widely used classification techniques including multilayer perceptron (MLP) neural networks, support vector machines (SVM), and decision trees (DT) based on two well-known combination methods including bagging and boosting and different numbers of combined classifiers. Our experimental results by three public datasets show that DT ensembles composed of 80–100 classifiers using the boosting method perform best. The Wilcoxon signed ranked test also demonstrates that DT ensembles by boosting perform significantly different from the other classifier ensembles. Moreover, a further study over a real-world case by a Taiwan bankruptcy dataset was conducted, which also demonstrates the superiority of DT ensembles by boosting over the others.

1. Introduction

Developing an effective bankruptcy prediction model is a very important but rather difficult task for financial institutions. The aim of bankruptcy prediction models is to predict whether or not a new applicant (including individual and company) will go bankruptcy or not. If the prediction models could not perform well (i.e. to provide a certain, high prediction error rate) it will lead to make incorrect decisions and hence, very likely to cause great financial crises and distress [29].

Similar to the objective of bankruptcy prediction, credit scoring (or rating) focuses on determining if loan customers belong to either a good or a bad applicant group. In other words, an effective credit scoring model can also help financial instructions decide whether or not to grant a credit to new applicants [10]. Particularly, both bankruptcy prediction and credit scoring are regarded as the financial decision making problems as well as binary classification problems. That is, the model is designed to assign new observations to two pre-defined classes, which are 'good' and 'bad' risk classes [26]. That is, if a credit scoring model classifies a new observation into the 'bad' risk class, this is similar to a bankruptcy prediction model that forecasts the new observation to be bankrupt. In other words, a 'bad' risk case can be simply regarded as the same as the 'bankruptcy' case.

Related literature and studies have shown that machine learning techniques, such as neural networks outperform conventional statistical techniques including logistic regression, in terms of prediction accuracy and error [27,29]. In specific, combining multiple classification techniques or classifier ensembles perform far better than single classification techniques [17].

Generally speaking, classifier ensembles are based on training a fixed number of classifiers for the same domain problems (or the training sets), and the final output over a given unknown data sample can be obtained by combining the outputs made by the trained classifiers. In literature, bagging and boosting are the two widely used combination methods [17] (c.f. Section 3.2).

Although many related studies have demonstrated the superiority of classifier ensembles over many single classifiers, most of them only constructed a specific type of classifier ensembles for bankruptcy prediction, such as neural network ensembles [13,29,31,33] and decision tree ensembles [1,26,32,35]. In addition, most of these classifier ensembles are only based on one specific combination method, i.e. either bagging or boosting (c.f. Section 3.3).

Despite some previous works focus on comparing bagging and boosting methods [5,19], where their findings show that the boosting method outperforms the bagging method, they conclude that the performances of classifier ensembles by bagging and boosting are usually domain dependent.

Therefore, in the domain problems of bankruptcy prediction and credit scoring assessment there is no comparative study to assess the performances of a good

[*] Corresponding author. Tel.: +1 607 436 3458; fax: +1 607 436 2543.
E-mail addresses: cftsai@mgt.ncu.edu.tw (C.-F. Tsai), d974020002@student.nsysu.edu.tw (Y.-F. Hsu), David.Yen@oneonta.edu (D.C. Yen).
[1] Tel.: +886 3 422 7151; fax: +886 3 4254604.
[2] Tel.: +886 7 525 2000; fax: +886 7 5254799.

http://dx.doi.org/10.1016/j.asoc.2014.08.047
1568-4946/© 2014 Elsevier B.V. All rights reserved.

collection of different classifier ensembles. In other words, this fact raises our research question concerning which classifier ensembles perform best.

To construct classifier ensembles, three issues in general, need to be carefully addressed/examined. First of all, since there are various classification techniques available, which one can be the best technique for the construction of classifier ensembles? Secondly, how many classifiers should be combined in order to provide a better performance? Thirdly and finally, which combination method should be used to combine multiple outputs produced by individual classifiers for a final output? To take care of these three issues, it is critical to investigate how to construct the optimal classifier ensemble for bankruptcy prediction and credit scoring. More specifically, in addition to using single classifiers as the baseline classifiers, we can further identify the representative baseline of classifier ensembles for future research.

This paper is organized as follows. Section 2 overviews the basic concept of classifier ensembles followed after the introduction section. Section 3 discusses the critical issues of constructing classifier ensembles and then, provides a review of related works in this subject area. Section 4 presents the experimental results and the conclusion is provided in Section 5.

2. Classifier ensembles

In the areas of pattern recognition and machine learning, the combination of a number of classifiers has recently been a popular research direction [20,22,23]. Further, this combination approach can be regarded as either ensemble classifiers or modular classifiers. Ensemble classifiers aim at obtaining highly accurate classifiers by combining less accurate ones. They are basically proposed to improve the classification performance of a single classifier [14]. That is, the combination one is able to complement the errors made by the individual classifiers on different parts of the input space. From the above discussion, the performance of modular classifiers is likely to perform better than the one of the best single classifiers used in isolation.

The concept is further, inspired by the nature of information processing in the brain which is modular. That is, individual functions can be subdivided into functionally different subprocess or subtasks without mutual interference [8]. This forms the divide-and-conquer principle that a complex problem can be divided into subproblems (i.e. simpler task), which can then be resolved with a different neural net architecture or algorithm. Then, the ultimate solution is reassembled from the results of the subtasks [25].

In addition to accuracy improvement (i.e. better generalization), efficiency (i.e. learning speed) is another important advantage in combining classifiers since the modularity results in an architecture with a lesser complexity. Moreover, it is relative easier and faster to train the set of simpler functions. Modular architectures have also found to be favorable over a single model in terms of such advantages as interpretable representation, scaling and ease of modification of architecture [12].

Fig. 1 shows the general architecture of a classifier ensemble [9]. A number of differently classifiers (i.e. experts) share the input and whose outputs are combined to produce an overall output. Note that the experts can be trained by providing different examples (or different features) of a given training set or different learning models trained by the same training set.

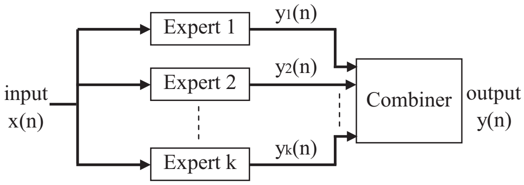

Fig. 1. Architecture of a classifier ensemble.

3. Issues related to developing classifier ensembles

3.1. Classification techniques

Bankruptcy prediction and credit scoring assessment can be approached by designating a single classifier. According to the study of Lin et al. [17], neural networks (especially multilayer perceptron networks), support vector machines, and decision trees are three most popular supervised learning techniques. These techniques are briefly introduced below.

3.1.1. Neural networks

Neural networks (or artificial neural networks) contain information-processing units similar to the neurons available in the human brain except that the information-processing units in a neural network are artificial [9]. Neural networks can learn by experience, generalize from previous experiences to new ones, and hence make useful decisions. A neural network consists of neural nodes which are linked to weighted nodes. Nodes and connections among nodes are analogous to brain neurons and synapses connecting brain neurons, respectively.

The most common neural network model is the multilayer perceptron (MLP) network, which includes an input layer with a set of sensory nodes as input nodes, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input nodes/neurons are the feature values of an instance whereas the output nodes/neurons are discriminators between the class of the instance and those of all other instances.

According to the study of Haykin [9], input vector x in a multilayer architecture passes through the network via the hidden layer of neurons to the output layer. The weight connecting input element i to hidden neuron j is denoted by W_{ji}, and the weight connecting hidden neuron j to output neuron k is denoted by V_{kj}. The net input of a neuron can be calculating by determining the weighted sum of its inputs while its output can be determined by a sigmoid function. Therefore, for the jth hidden neuron

$$net_j^h = \sum_{i=1}^{N} W_{ji} x_i \text{ and } y_i = f(net_j^h) \tag{1}$$

while for the kth output neuron

$$net_k^o = \sum_{j=1}^{J+1} V_{kj} y_i \text{ and } o_k = f(net_k^o) \tag{2}$$

The sigmoid function $f(net)$ is the logistic function

$$f(net) = \frac{1}{1+e^{-\lambda net}} \tag{3}$$

where λ controls the gradient of the function.

For a given input vector, the network produces an output o_k. Each response is then compared to the known desired response of each neuron d_k. All weights in the network are then, modified continuously to correct and/or reduce errors until the total error from all training examples is limited to a pre-defined tolerance level.

For the output layer weights V and the hidden layer weights W, the update rules are given in Eqs. (4) and (5), respectively

$$V_{kj}(t+1) = v_{kj}(t) + c\lambda(d_k - o_k)o_k(1-o_k)y_j(t) \tag{4}$$

$$W_{ji}(t+1) = w_{ji}(t) + c\lambda^2 y_j(1-y_j)x_i(t)\left(\sum_{k=1}^{K}(d_k - o_k)o_k(1-o_k)v_{kj}\right) \tag{5}$$