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/

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.

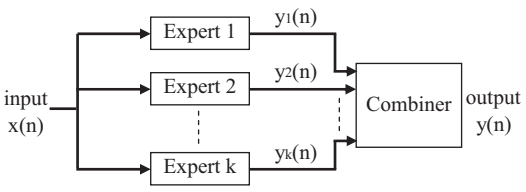

Fig. 1. Architecture of a classifier ensemble.

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

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

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

3.1.2. Support vector machines

Vapnik [31] first introduced support vector machines (SVMs) to perform binary classification – i.e., to separate a set of training vectors for two different classes $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$ where $x_i \in R^d$ denotes vectors in a d-dimensional feature space, and $y_i \in \{-1, +1\}$ is a class label. To generate an SVM model, input vectors are mapped onto a new higher dimensional feature space denoted as $\Phi : R^d \rightarrow H^f$ where $d < f$. An optimal separating hyperplane in the new feature space is then constructed by a kernel function $K(x_i, x_j)$, which is the product of input vectors x_i and x_j and where $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$.

3.1.3. Decision trees

The decision tree takes the form of a top-down tree structure, which splits the data to create leaves. A decision tree is built where each internal node denotes a test on an attribute and each branch represents an outcome of the test. The leaf nodes represent either classes or class distributions. The top-most node in a tree is the root node with the highest information gain. After the root node, one of the remaining attribute with the highest information gain is then chosen as the test for the next node. This process continues until all the attributes are compared or there are no remaining attributes on which the samples may be further partitioned [3].

The attribute with the highest information gain (or greatest entropy reduction) is chosen as the test attribute for the current node. Such an information-theoretic approach minimizes the expected number of tests needed to classify an object and guarantees that a simple (but not necessarily the simplest) tree is found.

Imagine selecting one case at random from a set S of cases and announcing that it belongs to some class C_j. The probability that an arbitrary sample belongs to class C_j is estimated by

$$P_i = \frac{freq(C_j, S)}{|S|} \tag{6}$$

where $|S|$ denotes the number of samples in the set S, and so the information it conveys is $-\log_2 p_i$ bits.

Suppose a probability distribution $P = \{p_1, p_2, \ldots, p_n\}$ is given then the information conveyed by this distribution, also called the entropy of P, is well known as

$$Info(P) = \sum_{i=1}^{n} - p_i \log_2 p_i \tag{7}$$

If we partition a set T of samples based on the value of a non-categorical attribute X into sets $T_1, T_2, \ldots, T_m$, then the information needed to identify the class of an element of T becomes the weighted average of the information needed to identify the class of an element of T_i, i.e. the weighted average of $Info(T_i)$

$$Info(X, T) = \sum_{i=1}^{m} \frac{|T_i|}{T} \times Info(T_i) \tag{8}$$

The information gain, $Gain(X, T)$, is then defined as

$$Gain(X, T) = Info(T) - Info(X, T) \tag{9}$$

This represents the difference between the information needed to identify an element of T and the information needed to identify an element of T after the value of attribute X has been evaluated. Thus, it is the gain in information due to attribute X.

3.2. Combination methods

3.2.1. Voting

The simplest method to combine classifiers is actually majority voting. The binary outputs of the k individual classifiers are pooled together. Then, the class which receives the largest number of votes is selected as the final classification decision [14]. In general, the final classification decision that reaches the majority of $(k+1)/2$ votes is taken.

3.2.2. Bagging

In bagging, several classifiers are trained independently by different training sets via the bootstrap method [2]. Bootstrapping builds k replicate training data sets to construct k independent networks by randomly re-sampling the original given training dataset, but with replacement. That is, each training example may appear to be repetitive but not at all in any particular replicate training data set of k. Then, the k networks are aggregated via an appropriate combination method such as majority voting.

3.2.3. Boosting

In boosting, similar to bagging, each classifier is trained using a different training set. However, the k networks are trained not in a parallel and independent way, but sequentially instead. The original boosting approach, *boosting by filtering*, was proposed by Schapire [24]. In boosting by filtering, three experts are individually trained. The first one is trained on m training examples. Then, the second expert is also trained by the m training examples. These training data are selected from the pool of the training data such that half of them are classified correctly and half of them are classified incorrectly by the first expert. Therefore, the second expert obtains 50% of patterns for training which were misclassified by expert one. Next, the third expert is trained only on data that the first two experts disagree. The classification decision is made by a majority vote of these three experts. However, this method requires a very large training data set to obtain m that expert one and two disagree.

AdaBoost is a combination of the ideas behind boosting and bagging and does not demand a large training data set as the other two. Initially, each training example of a given training set has the same weight. For training the kth classifier as a *weak learning model*, n sets of training samples $(n < m)$ among S are used to train the kth classifier. Then, the trained classifier is evaluated by S to identify those training examples which cannot be classified correctly. The $k+1$ network is then trained by a modified training set which boosts the importance of those incorrectly classified examples. This sampling procedure will be repeated until K training samples is built for constructing the kth network. Therefore, the final decision is based on the weighted vote of the individual classifiers [5,6].

Note that for the theoretical comparison between different combination methods, please refer to [7,16].

3.3. Comparisons of related work

Table 1 lists these prior studies that had developed classifier ensembles by using neural networks, support vector machines, and decision trees. In addition, several attributed including the combination methods used, the number of combined classifiers, the baseline classifiers compared, and the datasets used for experiments are used to compare and contrast the relative differences.

According to Table 1, it is noted that constructing classifier ensembles is an active research area in the areas of bankruptcy prediction and credit scoring where bagging and boosting are widely used combination methods. Related studies applying classifier ensemble techniques have shown that they are superior to many single classification techniques. In literature, there are several works that compare different classifier ensembles. For example, Dietterich [5] construct decision tree ensembles by bagging, boosting, and randomization. The findings are that the better performance depends on the level of classification noise. On the other hand, Opitz and Maclin [19] compare neural network ensembles and decision tree ensembles by bagging and boosting over 23

Table 1
Comparisons of related works.

Work	Classification technique	Combination method	No. of combined classifiers	Baseline	Dataset
Wang et al. [32]	DT ensembles	Bagging	10, 50, 100, 150	Random forest/LR[a]/LDA[b]/MLP/RBFN[c]	Australian/German
Sun et al. [26]	DT ensembles	Adaboost	10	DT/SVM	Chinese listed companies
Kim and Kang [13]	MLP ensembles	Bagging/Boosting	10	MLP/DT	Korean listed companies
Paleologo et al. [21]	DT ensembles/SVM ensembles	Bagging	20–50	DT/SVM/*k*-NN[d]	IBM Italian clients
Twala [30]	MLP ensembles/DT ensembles	Bagging/Boosting	5	MLP/DT/LR/Naive Bayes	Australian/German
Zhang et al. [35]	DT ensembles	Bagging	10, 20, 30 to 120	DT/MLP/SVM/NN ensembles	Australian/German
Nanni and Lumini [18]	MLP ensembles/SVM ensembles/*k*-NN ensembles	Bagging	50	MLP/SVM/*k*-NN	Australian/German/Japanese
Alfaro et al. [1]	DT ensembles	Adaboost	100	LDA/MLP/DT	SABI database of Bureau Van Dijk
Tsai and Wu [29]	MLP ensembles	Voting	3, 5, 7, 9, 11, 12, 15	MLP	Australian/German/Japanese
West et al. [34]	MLP ensembles	Bagging/Boosting	5, 10, 15, 20 to 100	MLP	Australian/German
West [33]	MLP ensembles	Mixture of experts [7]	6	MLP/LDA/LR/DT/RBFN	Australian/German

[a] LR: logistic regression.
[b] LDA: linear discriminant analysis.
[c] RBFN: radial basis function network.
[d] *k*-NN: *k*-nearest neighbor.

datasets. They observe that bagging is sometimes much less accurate than boosting, but the performance of the boosting method is dependent on the characteristics of the dataset being examined.

However, for the bankruptcy prediction problem, most studies had constructed specific classifier ensembles without considering different classifier ensembles for a comparison purpose (c.f. Table 1). In addition, previous works of comparing different classifier ensembles do not provide clues for constructing the best bankruptcy prediction model by classifier ensembles. Therefore, there is no clear answer to the question about which classifier ensemble can provide the highest rate of prediction accuracy.

More specifically, since bagging and boosting/Adaboost are two most widely used combination methods to construct classifier ensembles, many of these aforementioned studies only apply one of them (i.e. either bagging or boosting/Adaboost). In other words, very few studies examine the performances of different classifier ensembles by bagging and boosting respectively.

Finally, the numbers of combined classifiers considered in related studies are quite different. That is, some combine a fixed number of multiple classifiers (e.g. 10) while some others use different numbers of combined classifiers for comparisons. This again leads to an urgent need of conducting a comprehensive study of constructing classifier ensembles by different combination methods with different numbers of multiple classifiers for bankruptcy prediction and credit scoring.

4. Experiments

4.1. Experimental setup

To make a comprehensive comparison, three related datasets are chosen and they are Australian,[3] German,[4] and Japanese[5] credit datasets. In particular, these datasets contain different numbers of variables and data samples (c.f. Table 2). Moreover, in this paper a 10-fold cross-validation method is used [15]. It is basically based on dividing ten equal parts of a dataset. Any nine of the ten subsets (or segments) are selected to perform classifier training. The remaining part will be executed for testing the classifier. As a result, each part will be trained and tested ten times.

Table 2
The dataset information.

	Total cases	Good/bad cases	No. of attributes
Australian credit	690	307/383	14
German credit	1000	700/300	20
Japanese credit	690	307/383	15

For classifier design, since neural networks, support vector machines, and decision trees [17] are three most widely used classification techniques, they are constructed as the single baseline classifiers. The parameter settings for multilayer perceptron (MLP) networks and support vector machines (SVM) are based on the findings of Tsai [24]. In addition, for the decision tree (DT) classifier, J48 and CART (Classification and Regression Tree) were compared and we found that the decision tree based on J48 performs relatively better.[6] Therefore, we only report the results of the DT classifier ensembles by J48.

Similarly, for classifier ensembles Table 1 also shows that most related studies construct classifier ensembles based on three classification techniques. More specifically, boosting and bagging are two widely applied combination methods. Therefore, six different classifier ensembles are thus, constructed and compared, which are MLP ensembles by boosting and bagging, SVM ensembles by boosting and bagging, and DT ensembles by boosting and bagging. Furthermore, we also compare different numbers of combined classifiers, which are 10, 20, 30, . . ., and 100 in order to find out the best classifier ensembles. Note that the WEKA (Waikato Environment for Knowledge Analysis) suite [11] was used to conduct our experiments.[7]

[3] http://www.liacc.up.pt/ML/statlog/datasets/australian/australian.doc.html.
[4] http://www.liacc.up.pt/ML/statlog/datasets/german/german.doc.html.
[5] http://www.ics.uci.edu/~mlearn/MLRepository.html.

[6] For the Australian, German, and Japanese datasets, J48 provides 86.07%, 73.37%, and 86.37% respectively and CART provides 85.92%, 73.77%, and 84.84% respectively.
[7] The followings show the process of constructing the classifier ensembles by boosting/bagging: "Classify" → "Choose" → "classifiers" → "meta" → "AdaBoostM1/bagging"; the numbers of combined classifiers are based on the setting of "numIterations".

4.2. Experimental results

4.2.1. Results on classifier ensembles

Figs. 2–4 show the prediction performances of the classifier ensembles over the Australian, German, and Japanese datasets respectively. Note that the *x*-axis and *y*-axis represent the number of combined classifiers and the rate of prediction accuracy.

On average, DT ensembles by boosting perform best over the Australian and Japanese datasets, which are 86.68% and 87.66% respectively. For the German dataset, SVM by boosting outperforms the others (i.e. 75.78%). However, the best performances (i.e. the highest rate of prediction accuracy) over the Australian, German, and Japanese datasets are DT-boosting-100 (87.23%), MLP-bagging-100 (76.48%), and DT-boosting-60 (88.36%) respectively.

On the other hand, we can see that DT ensembles perform very differently over the German dataset, which are much more unstable than the others. One reason for this phenomenon may be due to the fact that German dataset size is the largest one if compared with the other two datasets as it contains the largest numbers of variables (i.e. 20) and data samples (1000). This finding may also imply that the performance of DT ensembles by boosting may be affected by larger datasets. In addition, for better prediction performance larger numbers of combined DT classifiers are needed. That is, only combine 100 DT classifiers by boosting and bagging can reach to the second highest rate of prediction accuracy (i.e. 75.98%).

Opposed to DT ensembles, SVM-boosting and SVM-bagging perform very stable no matter how many numbers of SVM classifiers are combined. Particularly, on average SVM-boosting and SVM-bagging perform the best and the second best over the German dataset. This fact may imply that SVM ensembles are good at handling larger scale datasets. In addition, although SVM ensembles do not provide a better performance than DT ensembles over the Australian and Japanese datasets on average, their performances are not heavily dependent on the numbers of combined classifiers.

For MLP ensembles by boosting, they perform worst over the Australian and Japanese datasets and second worst over the German dataset. However, MLP ensembles by bagging perform second best over the Japanese dataset and they can even provide the highest accuracy rate over the German dataset when 100 MLP classifiers are combined.

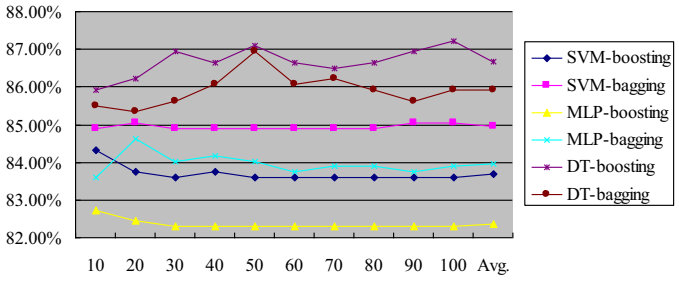

Fig. 2. Classification accuracy over the Australian dataset.

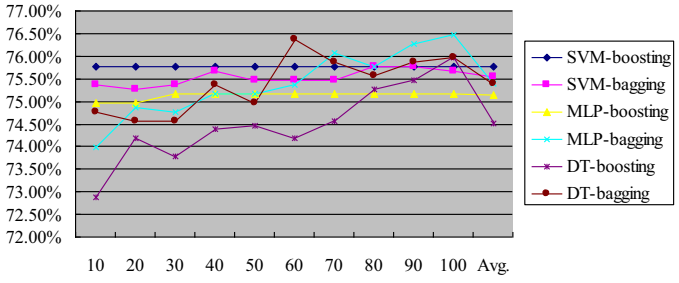

Fig. 3. Classification accuracy over the German dataset.

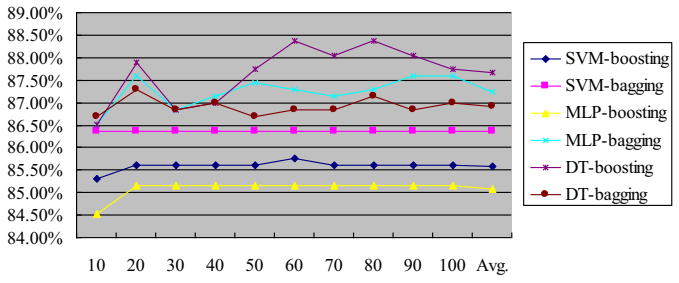

Fig. 4. Classification accuracy over the Japanese dataset.

4.2.2. Comparisons between classifier ensembles and single classifiers

Table 3 shows the performance differences between the single classifiers and the best classifier ensembles over the three different datasets. For classifier ensembles, the value in brackets following accuracy means the combination method and numbers of combined classifiers,[8] and another value in brackets means the performance ranking over a specific dataset.

These results indicate that DT ensembles by boosting can be regarded as the optimal classifier ensemble technique for bankruptcy prediction. Specifically, they can provide the highest rate of accuracy over the Australian and Japanese datasets. For the German dataset, both DT ensembles by boosting and bagging perform second best but the accuracy rate is slightly lower than MLP ensembles by bagging. If we compare single DT classifier and DT ensembles, the improvement by the classifier ensemble technique is rather significant. One limitation of DT ensembles is that they need to combine larger numbers of DT classifiers in order to provide optimal performance.

The performance differences between SVM and SVM ensembles over the three datasets are however, very small. In particular, SVM ensembles by boosting and bagging cannot improve the prediction performance when compared with SVM alone. These results show that SVM and SVM ensembles may not be the best solution for bankruptcy prediction.

On the other hand, the performance by MLP ensembles is certainly improved when compared with MLP. However, similar to SVM and SVM ensembles, MLP and MLP ensembles are not the single best classifier and the best classifier ensembles.

Finally, to examine the level of significant difference between the six different types of classifier ensembles, Wilcoxon signed ranked test [4] is used. Tables 4–6 show the results of different classifier ensembles over the Australian, German, and Japanese datasets respectively.

These results indicate that DT ensembles by boosting perform significantly different from the other classifier ensembles over the Australian dataset. For the German dataset, most classifier ensembles do not perform significantly different. This implies that there is no exact winner over larger datasets. For the Japanese dataset, DT-boosting has a significant level of performance difference except for comparing with MLP bagging, which performs the second highest rate of prediction accuracy.

4.2.3. A case study

In addition to using the three simulation datasets, we further examine the prediction performances of the constructed classifier ensembles over a real-world dataset to see their applicability to the real-world problem. The data samples were collected from the

[8] If the classifier ensembles by different numbers of combined classifiers perform the same, we only present the smallest numbers of combined classifiers in Table 2.

Table 3
Comparisons between single classifiers and their best classifier ensembles.

	Australian	German	Japanese
SVM	85.63% (2)	75.68% (4)	86.37% (3)
SVM ensembles	85.05% (bagging-90) (3)	75.78% (boosting-10) (3)	86.37% (bagging-10) (3)
MLP	82.44% (6)	70.57% (6)	84.38% (6)
MLP ensembles	84.62% (bagging-20) (5)	76.48% (bagging-100) (1)	87.60% (bagging-20) (2)
DT	84.91% (4)	73.77% (5)	86.37% (5)
DT ensembles	87.23% (boosting-100) (1)	75.98% (boosting/bagging-100) (2)	88.36% (boosting-80) (1)

Table 4
Wilcoxon signed ranked test over the Australian dataset (p value).

	SVM-bagging	MLP-boosting	MLP-bagging	DT-boosting	DT-Bagging
SVM-boosting	0.004	0.004	0.058	0.005	0.005
SVM-bagging		0.004	0.005	0.005	0.005
MLP-boosting			0.005	0.005	0.005
MLP-bagging				0.005	0.005
DT-boosting					0.005

Table 5
Wilcoxon signed ranked test over the German dataset (p value).

	SVM-bagging	MLP-boosting	MLP-bagging	DT-boosting	DT-Bagging
SVM-boosting	0.011	0.003	0.138	0.007	0.126
SVM-bagging		0.005	0.635	0.008	0/386
MLP-boosting			0.263	0.059	0.167
MLP-bagging				0.005	0.570
DT-boosting					0.008

Table 6
Wilcoxon signed ranked test over the Japanese dataset (p value).

	SVM-bagging	MLP-boosting	MLP-bagging	DT-boosting	DT-Bagging
SVM-boosting	0.003	0.003	0.005	0.005	0.005
SVM-bagging		0.002	0.005	0.005	0.005
MLP-boosting			0.005	0.005	0.005
MLP-bagging				0.02	0.015
DT-boosting					0.017

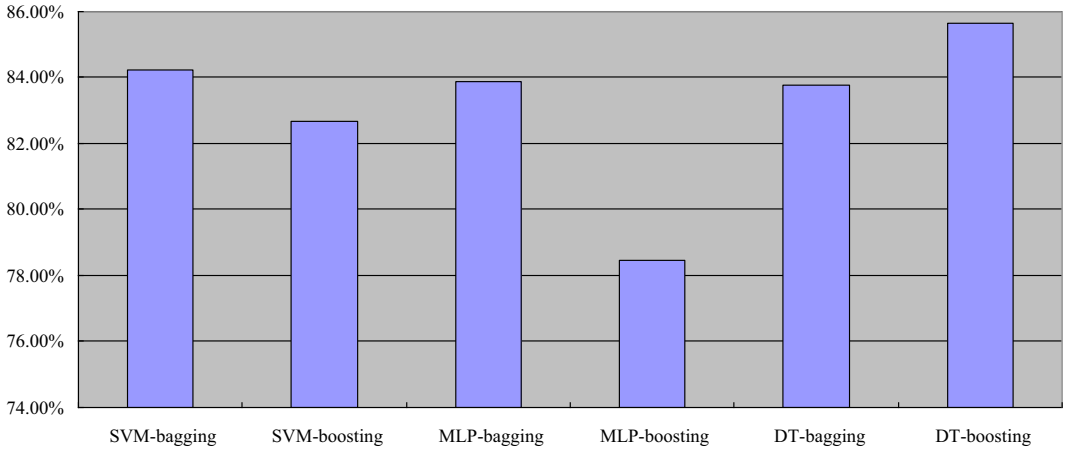

Fig. 5. Prediction performances of SVM ensembles, MLP ensembles, and DT ensembles.

Taiwan Economic Journal[9] and the definitions of bankrupt companies are based on the business regulations from the Taiwan Stock Exchange. As a result, the dataset is composed of 220 good and 220 bad cases respectively. In addition, each sample contains 95 different attributes (i.e. relevant financial ratios). Fig. 5 shows the prediction results by SVM ensembles (i.e. bagging-90 and boosting-10), MLP ensembles (i.e. bagging-100 and boosting-20), and DT ensembles (i.e. bagging-100 and boosting-100). This result is consistent with the previous finding that the DT ensemble by boosting outperforms the other classifier ensembles, which provide 85.65% prediction accuracy.

In addition to prediction accuracy, the computational costs of training these classifier ensembles are also compared (c.f. Table 7). The result shows that DT ensembles by boosting require relatively low computational cost during classifier training, which is the top three efficient classifier ensembles.

[9] http://www.tej.com.tw/twsite/.

Table 7
The times of training different classifier ensembles (seconds).

Classifier ensembles	Training time
SVM-boosting-10	1.48
SVM-bagging-90	12.84
MLP-boosting-20	27.75
MLP-bagging-100	214.6
DT-boosting-100	11.11
DT-bagging-100	9.54

In summary, DT ensembles by boosting mostly outperform the other ensembles. That is, combining 100 DT classifiers by the boosting combination method can provide the best results over the Australia, Japanese, and Taiwanese datasets. Particularly, the results of combining different numbers of DT classifiers are much more different than the other ensembles. In other words, although DT ensembles by boosting can be regarded as a representative classifier ensemble for financial distress prediction, the number of combined DT classifiers should be carefully designed. On the other hand, MLP ensembles and SVM ensembles perform more stable than DT ensembles when different numbers of classifiers are combined.

Despite the best DT ensembles should be based on combining a large number of DT classifiers, which is larger than the other ensembles, the computational cost of constructing the best DT ensembles is not very high. This is because MLP and SVM per se is more complex to design, which require larger times for developing the MLP and SVM classifiers than a DT classifier.

5. Conclusion

Since the performances of classifier ensembles in bankruptcy prediction have not fully examined in literature, the aim of this paper is to conduct a comprehensive study of comparing well-known and widely developed classifier ensembles in terms of bankruptcy prediction and credit scoring. More specifically, there are two research questions that we attempt to answer. First, as bagging and boosting are the two most popular methods to combine multiple classifiers, it is not known which combination method by what classification techniques performs the best? Second, since classifier ensembles are based on combining a number of specific classifiers, how many classifiers should be combined in order to provide the best performance?

To answer these two research questions, three widely used classification techniques including multilayer perceptron (MLP) neural networks, support vector machines (SVM), and decision trees (DT) are taken into account. In addition, MLP ensemble, SVM ensembles, and DT ensembles by bagging and boosting are constructed for performance comparison. Our experimental results based on three related public datasets show that DT ensembles by boosting perform best, which outperform the other types of classifier ensembles and single classifiers. Moreover, a real-world case by a Taiwan bankruptcy dataset was used for further comparison, where the result demonstrates the outperformance of DT ensembles by boosting over the others ensembles. On the other hand, the average computational cost of training DT ensembles by boosting is relatively low, which is more efficient than SVM ensembles by bagging, MLP ensembles by boosting, and MLP ensembles by bagging. Therefore, DT ensemble by boosting can be regarded as the baseline classifier ensemble technique in future related studies.

For future work, some other issues could also be considered. First of all, since there is no ground truth answer to the most representative features (i.e. input variables), some feature selection methods can be employed to study the effect of performing feature selection on classifier ensembles [28]. Secondly, in addition to assessing the performances of different single classifiers and classifier ensembles, hybrid classifiers which are based on combining cluster analysis as the first component and one specific classification technique for the second component can be further compared and contrasted to provide some additional insights [14]. Thirdly, in addition to comparing homogeneous classifier ensembles, which combine the same types of multiple classifiers, as this paper did, it would be valuable to further examine the performance of heterogeneous classifier ensembles, such as combining MLP, DT, and SVM by bagging and boosting for bankruptcy prediction and credit scoring.

References

[1] E. Alfaro, N. Garcia, M. Gamez, D. Elizondo, Bankruptcy forecasting: an empirical comparison of AdaBoost and neural networks, Decis. Support Syst. 45 (1) (2008) 110–122.
[2] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[3] L. Breiman, J.H. Friedman, R.A. Olshen, P.J. Stone, Classification and Regressing Trees, Wadsworth International Group, California, 1984.
[4] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (December) (2006) 1–30.
[5] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (2000) 139–157.
[6] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Proceedings of the International Conference on Machine Learning, Bari, Italy, July 3–6, 1996, pp. 148–156.
[7] G. Fumera, F. Roli, A theoretical and experimental analysis of linear combiners for multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 27 (6) (2005) 942–956.
[8] B.L.M. Happel, J.M.J. Murre, The design and evolution of modular neural network architectures, Neural Netw. 7 (6-7) (1994) 985–1004.
[9] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed., Prentice Hall, 1999.
[10] Z. Huang, H. Chen, C.-J. Hsu, W.-H. Chen, S. Wu, Credit rating analysis with support vector machines and neural networks: a market comparative study, Decis. Support Syst. 37 (4) (2004) 543–558.
[11] W.H. Ian, F. Eibe, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 1999.
[12] R.A. Jacobs, M.I. Jordan, A.G. Barto, Task decomposition through competition in a modular connectionist architecture: what and where vision tasks, Cognit. Sci. 15 (2) (1991) 219–250.
[13] M.-J. Kim, D.-K. Kang, Ensemble with neural networks for bankruptcy prediction, Expert Syst. Appl. 37 (4) (2010) 3373–3379.
[14] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[15] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, Int. Jt. Conf. Artif. Intell. (1995) 1137–1143.
[16] L.I. Kuncheva, A theoretical study oon six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 281–286.
[17] W.-Y. Lin, Y.-H. Hu, C.-F. Tsai, Machine learning in financial crisis prediction: a survey, IEEE Trans. Syst. Man Cybern. C: Appl. Rev. (2012), http://dx.doi.org/10.1109/TSMCC.2011.2170420.
[18] L. Nanni, A. Lumini, An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring, Expert Syst. Appl. 36 (2) (2009) 3028–3033.
[19] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198.
[20] N.C. Oza, K. Tumer, Classifier ensembles: select real-world applications, Inf. Fus. 9 (1) (2008) 4–20.
[21] G. Paleologo, A. Elisseeff, G. Antonini, Subagging for credit scoring models, Eur. J. Oper. Res. 201 (2) (2010) 490–499.
[22] R. Ranawana, V. Palade, Multi-classifier systems: review and a roadmap for developers, Int. Hybrid Intell. Syst. 3 (1) (2006) 35–61.
[23] L. Rokach, Taxonomy for characterizing ensemble methods in classification tasks: a review and annotated bibliography, Comput. Stat. Data Anal. 53 (12) (2009) 4046–4072.
[24] R.E. Schapire, The strength of weak learnabilty, Mach. Learn. 5 (2) (1990) 197–227.
[25] A.J.C. Sharkey, Combing Artificial Neural Nets: Ensemble and Modular Multi-Net Systems, Springer-Verlag, 1999.
[26] J. Sun, M.-Y. Jia, H. Li, AdaBoost ensemble for financial distress prediction: an empirical comparison with data from Chinese listed companies, Expert Syst. Appl. 38 (8) (2011) 9305–9312.
[27] C.-F. Tsai, Financial decision support using neural networks and support vector machines, Exp. Syst. 25 (4) (2008) 380–393.
[28] C.-F. Tsai, Feature selection in bankruptcy prediction, Knowl. Based Syst. 22 (2) (2009) 120–127.
[29] C.-F. Tsai, J.-W. Wu, Using neural network ensembles for bankruptcy prediction and credit scoring, Expert Syst. Appl. 34 (4) (2008) 2639–2649.
[30] B. Twala, Multiple classifier application to credit risk assessment, Expert Syst. Appl. 37 (4) (2010) 3326–3336.

[31] V. Vapnik, Statistical Learning Theory, John Wiley, New York, 1998.
[32] G. Wang, J. Ma, L. Huang, K. Xu, Two credit scoring models based on dual strategy ensemble trees, Knowl. Based Syst. 26 (February) (2012) 61–68.
[33] D. West, Neural network credit scoring models, Comput. Oper. Res. 27 (11/12) (2000) 1131–1152.
[34] D. West, S. Dellana, J. Qian, Neural network ensemble strategies for financial decision applications, Comput. Oper. Res. 32 (10) (2005) 2543–2559.
[35] D. Zhang, X. Zhou, S.C.H. Leung, J. Zheng, Vertical bagging decision trees model for credit scoring, Expert Syst. Appl. 37 (2) (2010) 7838–7843.