Expert Systems with Applications 39 (2012) 10244–10250

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Exploring the behaviour of base classifiers in credit scoring ensembles

A.I. Marqués [a], V. García [b], J.S. Sánchez [b,*]

[a] Department of Business Administration and Marketing, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain
[b] Department of Computer Languages and Systems, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain

ARTICLE INFO

Keywords:
Finance
Credit scoring
Classifier ensemble

ABSTRACT

Many techniques have been proposed for credit risk assessment, from statistical models to artificial intelligence methods. During the last few years, different approaches to classifier ensembles have successfully been applied to credit scoring problems, demonstrating to be more accurate than single prediction models. However, it is still a question what base classifiers should be employed in each ensemble in order to achieve the highest performance. Accordingly, the present paper evaluates the performance of seven individual prediction techniques when used as members of five different ensemble methods. The ultimate aim of this study is to suggest appropriate classifiers for each ensemble approach in the context of credit scoring. The experimental results and statistical tests show that the C4.5 decision tree constitutes the best solution for most ensemble methods, closely followed by the multilayer perceptron neural network and logistic regression, whereas the nearest neighbour and the naive Bayes classifiers appear to be significantly the worst.

1. Introduction

The recent world financial crisis has aroused increasing attention of banks and financial institutions on credit risk. The main problem comes from the difficulty to distinguish the creditworthy applicants from those who will probably default on repayments. The decision to grant credit to an applicant was traditionally based upon subjective judgments made by human experts, using past experiences and some guiding principles. Common practice was to consider the classic five C's of credit: character, capacity, capital, collateral and conditions (Abrahams & Zhang, 2008). This method suffers, however, from high training costs, frequent incorrect decisions, and inconsistent decisions made by different experts for the same application.

These shortcomings have led to a rise in more formal and accurate methods to assess the risk of default. In this context, credit scoring and behavioural scoring have become primary tools for financial institutions to evaluate credit risk, improve cash flow, reduce possible risks and make managerial decisions (Thomas, Edelman, & Crook, 2002). Difference between credit scoring and behavioural scoring is that the former focuses on decisions regarding new applicants for credit, whereas the latter refers to monitoring and predicting the repayment behaviour of existing borrowers.

Credit scoring is essentially a set of techniques that help lenders decide whether or not to grant credit to applicants. The aim of credit scoring models is to discriminate between "good" and "bad" loans, depending on how likely applicants are to default with their repayments. Compared with the traditional subjective methods, credit scoring models present some advantages (Rosenberg & Gleit, 1994; Thomas et al., 2002): (i) they are cheaper to purchase and operate; (ii) they make faster credit decisions; (iii) they provide consistent recommendations based on objective information, thus eliminating human biases and prejudices; (iv) changes in policy and/or economy can easily be incorporated into the system; and (v) the performance of the credit scoring model can be monitored, tracked, and adjusted at any time.

From the seminal reference to credit scoring in the introductory paper by Altman (1968), many other developments have been subsequently proposed in the literature. The most classical approaches to credit scoring are based on statistical and mathematical programming models, such as linear and quadratic discriminant analysis, linear and logistic regression, multivariate adaptive regression splines, Markov chain models, and linear and quadratic programming. However, after the Basel II recommendations issued by the Basel Committee on Banking Supervision in 2004, financial institutions have required to use more complex credit scoring models in order for enhancing the efficiency of capital allocation.

In recent years, many studies have demonstrated that artificial intelligence techniques (decision trees, artificial neural networks, support vector machines, evolutionary computing) can be successfully used for credit evaluation (Chi & Hsu, 2012; Huang, Chen, Hsu, Chen, & Wu, 2004; Huang, Chen, & Wang, 2007; Ince & Aktan, 2009; Martens et al., 2010; Ong, Huang, & Tzeng, 2005). In contrast to the statistical models, the artificial intelligence methods do not assume any specific prior knowledge, but automatically extract information from past observations.

* Corresponding author. Tel.: +34 964 728350.
E-mail address: sanchez@uji.es (J.S. Sánchez).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.092

Although previous studies conclude that artificial intelligence techniques are superior to traditional statistical models, there is no overall best method for dealing with credit scoring problems. This is one of the main reasons why there exists an increasing interest in the use of classifier ensembles. Most of these works have demonstrated that the ensemble approach performs better than single classifiers when applied to credit scoring (Doumpos & Zopounidis, 2007; Hung & Chen, 2009; Twala, 2010; Wang, Hao, Ma, & Jiang, 2011; West, Dellana, & Qian, 2005). Despite the progress in this research field, several questions still remain open and should be addressed in order to fully understand the conditions under which the classifier ensembles can improve the model performance.

Taking these considerations into account, the present paper examines the use of seven well-known classifiers with five effective ensemble methods. The aim of this study is to find out what individual models are suitable for each ensemble strategy in the area of credit scoring. To this end, several experiments on six real credit data sets are carried out and the results are analysed for statistically significant differences by means of Friedman and Bonferroni-Dunn post hoc tests.

Hereafter, the paper is organized as follows. Section 2 gives a brief overview of the classifier ensemble approaches used in this study. Section 3 describes the set-up of the experiments carried out. Section 4 discusses the experimental results. Finally, Section 5 remarks the main findings and discusses future research directions.

2. Classifier ensembles

A classifier ensemble (also referred to as committee of learners, mixture of experts, multiple classifier system) consists of a set of individually trained classifiers (base classifiers) whose decisions are combined in some way, typically by weighted or unweighted voting, when classifying new examples (Kittler, 1998; Kuncheva, 2004). It has been found that in most cases the ensembles produce more accurate predictions than the base classifiers that make them up (Dietterich, 1997). Nonetheless, for an ensemble to achieve much better generalization capability than its members, it is critical that the ensemble consists of highly accurate base classifiers whose decisions are as diverse as possible (Bian & Wang, 2007; Kuncheva & Whitaker, 2003).

In statistical pattern recognition, a large number of methods have been developed for the construction of ensembles that can be applied to any base classifier. In the following sections, the ensemble approaches relevant for this study are briefly described.

2.1. Bagging

In its standard form, the bagging (Bootstrap Aggregating) algorithm (Breiman, 1996) creates M bootstrap samples $T_1, T_2, \ldots, T_M$ randomly drawn (with replacement) from the original training set T of size n. Each bootstrap sample T_i of size n is then used to train a base classifier C_i. Predictions on new observations are made by taking the majority vote of the ensemble C^* built from $C_1, C_2, \ldots, C_M$. As bagging resamples the training set with replacement, some instances may be represented multiple times while others may be left out.

Since each ensemble member is not exposed to the same set of instances, they are different from each other. By voting the predictions of each of these classifiers, bagging seeks to reduce the error due to variance of the base classifier.

2.2. Boosting

Similar to bagging, boosting also creates an ensemble of classifiers by resampling the original data set, which are then combined by majority voting. However, in boosting, resampling is directed to provide the most informative training data for each consecutive classifier.

The AdaBoost (Adaptive Boosting) algorithm proposed by Freund and Schapire (1996) constitutes the best known member in boosting family. It generates a sequence of base classifiers $C_1, C_2, \ldots, C_M$ by using successive bootstrap samples $T_1, T_2, \ldots, T_M$ that are obtained by weighting the training instances in M iterations. AdaBoost initially assigns equal weights to all training instances and in each iteration, it adjusts these weights based on the misclassifications made by the resulting base classifier. Thus, instances misclassified by model C_{i-1} are more likely to appear in the next bootstrap sample T_i. The final decision is then obtained through a weighted vote of the base classifiers (the weight w_i of each classifier C_i is computed according to its performance on the weighted sample T_i it was trained on).

2.3. Random subspace

The random subspace method is an ensemble construction technique proposed by Ho (1998), in which the base classifiers $C_1, C_2, \ldots, C_M$ are trained on data sets $T_1, T_2, \ldots, T_M$ constructed with a given proportion K of attributes picked randomly from the original set of features F. The outputs of the models are then combined, usually by a simple majority voting scheme.

This method may benefit from using random subspaces for both constructing and aggregating the classifiers. When the data set has many redundant attributes, one may obtain better classifiers in random subspaces than in the original feature space. The combined decision of such classifiers may be superior to a single classifier constructed on the original training data set in the complete feature space. On the other hand, when the number of training cases is relatively small compared with the data dimensionality, by constructing classifiers in random subspaces one may solve the small sample size problem.

2.4. DECORATE

Melville and Mooney (2005) introduced a new ensemble approach called DECORATE (Diverse Ensemble Creation by Oppositional Relabelling of Artificial Training Examples), which uses an existing learner to build an effective diverse committee in an iterative manner.

At each iteration, some artificial instances are randomly generated and combined with the original training data T in order to build a new ensemble member C_i. The labels for these artificially generated training instances are chosen so as to differ maximally from the current ensemble predictions, thereby increasing diversity when a new classifier is trained on the augmented data and added to the ensemble. While forcing diversity, it is still possible to maintain training accuracy by rejecting a new classifier if incorporating it into the existing ensemble decreases its performance.

2.5. Rotation forest

Rotation forest (Rodriguez, Kuncheva, & Alonso, 2006) refers to a technique to generate an ensemble of classifiers, in which each base classifier is trained with a different set of extracted attributes.

The main heuristic is to apply feature extraction and to subsequently reconstruct a full attribute set for each classifier in the ensemble. To this end, the feature set F is randomly split into K subsets, principal component analysis (PCA) is run separately on each subset, and a new set of linear extracted attributes is constructed by pooling all principal components. The data is transformed linearly into the new feature space. Classifier C_i is trained with this data set. Different splits of the feature set will lead to

different extracted features, thereby contributing to the diversity introduced by the bootstrap sampling.

3. Experiments

The aim of the experiments here carried out is to analyse the performance of five ensemble methods and investigate to what extent the behaviour of each technique is affected by the base classifier. The ensemble approaches here used are those described in Section 2: bagging, AdaBoost, random subspace, DECORATE and rotation forest. The base classifiers in each of these ensembles correspond to seven well-known models suitable for credit scoring: 1-nearest neighbour (1-NN), naive Bayes classifier (NBC), logistic regression (logR), multilayer perceptron (MLP) and radial basis function (RBF) neural networks, support vector machine (SVM) with a linear kernel, and C4.5 decision tree. In total, we have analysed the performance of 35 classifier ensembles for several credit scoring applications.

All classifiers have been implemented using the WEKA environment (Hall et al., 2009), which is a big collection of statistical and machine learning algorithms for preprocessing, classification and regression in data mining problems. Default parameter values in WEKA have been used for the base classifiers, which is a common practice in literature.

3.1. Description of the experimental databases

Six real-world credit data sets have been taken to compare the performance of the rotation forests with other classifier ensembles. The widely-used Australian, German and Japanese data sets are from the UCI Machine Learning Database Repository (http://archive.ics.uci.edu/ml/). The UCSD data set corresponds to a reduced version of a database used in the 2007 Data Mining Contest organized by the University of California San Diego and Fair Isaac Corporation. The Iranian data set comes from a modification to a corporate client database of a small private bank in Iran (Sabzevari, Soleymani, & Noorbakhsh, 2007). The Polish data set contains bankruptcy information of 120 companies recorded over a two-year period (Pietruszkiewicz, 2008). Table 1 reports a summary of the main characteristics of these benchmarking data sets.

3.2. Experimental protocol

The standard way to assess credit scoring systems is to employ a holdout sample since large sets of past applicants are usually available. However, there are situations in which data are too limited to build an accurate scorecard and therefore, other strategies have to be applied in order to obtain a good estimate of the classification performance. The most common way around this corresponds to cross-validation (Thomas et al., 2002, Ch. 7).

Accordingly, a 5-fold cross-validation method has been adopted for the present experiments: each original data set has been randomly divided into five stratified parts of equal (or approximately equal) size. For each fold, four blocks have been pooled as the training data, and the remaining part has been employed as an independent test set. Besides, ten repetitions have been run for each trial. The results from classifying the test samples have been averaged across the 50 runs and then evaluated for significant differences between models using the Friedman and Bonferroni-Dunn tests at significance levels of $\alpha = 0.05$ and $\alpha = 0.10$.

Table 1
Some characteristics of the data sets used in the experiments.

Data set	#Attributes	#Good	#Bad
Australian	14	307	383
German	24	700	300
Japanese	15	296	357
Iranian	27	950	50
Polish	30	128	112
UCSD	38	1836	599

3.3. Evaluation criteria

Standard performance evaluation criteria in the fields of credit scoring include accuracy, error rate, Gini coefficient, Kolmogorov–Smirnov statistic, mean squared error, area under the ROC curve, and type-I and type-II errors (Abdou & Pointon, 2011; Hand, 2005; Thomas et al., 2002; Yang, Wang, Bai, & Zhang, 2004). For a two-class problem, most of these metrics can be easily derived from a 2×2 confusion matrix as that given in Table 2, where each entry (i,j) contains the number of correct/incorrect predictions.

Most of credit scoring applications often utilize the accuracy rate (also called score of hits) as the criterion for performance evaluation. It represents the proportion of the correctly classified cases (both good and bad applicants) on a particular data set, and can be formally defined as follows:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d}$$

However, the accuracy ignores the different cost of both error types (bad applicants being predicted as good, or vice versa). This is the reason why it becomes especially interesting to measure the error on each individual class by using the type-I and type-II errors.

$$\text{Type-}I\text{error} = \frac{c}{c+d} \qquad \text{Type-}II\text{error} = \frac{b}{a+b}$$

Type-I error (or miss) is the rate of bad applicants being categorized as good. When this happens, the misclassified bad applicants will become default. Therefore, if the credit granting policy of a financial institution is too generous, this will be exposed to high credit risk. Type-II error (or false-alarm) defines the rate of good applicants being predicted as bad. When this happens, the misclassified good applicants are refused and therefore, the financial institution has opportunity cost caused by the loss of good customers. As stated by Caouette, Altman, Narayanan, and Nimmo (2008), the misclassification costs associated with type-I errors are typically much higher than those associated with type-II errors.

3.4. Statistical significance tests

Probably, the most common way to compare two or more classifiers over various data sets is the Student's paired t-test, which checks whether the average difference in their performance over the data sets is significantly different from zero. However, this appears to be conceptually inappropriate and statistically unsafe because parametric tests are based on a variety of assumptions (normality, large number of data sets, homogeneity of variance) that are often violated due to the nature of the problems (Demšar, 2006; García, Fernández, Luengo, & Herrera, 2010).

In general, the non-parametric tests (e.g., Wilcoxon and Friedman tests) should be preferred over the parametric ones because they do not assume normal distributions or homogeneity of variance. In this work, we have adopted the Friedman test to compare

Table 2
Confusion matrix for a credit scoring problem.

	Predicted as good	Predicted as bad
Good applicant	a	b
Bad applicant	c	d

Table 3
Accuracy rate and Friedman average ranking for the base classifiers.

	Australian	German	Japanese	Iranian	Polish	UCSD	Average rank
1-NN	81.45	70.50	79.48	93.00	75.42	80.04	5.17
NBC	80.72	74.90	82.08	23.20	68.33	67.72	6.00
logR	84.93	75.70	87.29	94.20	72.92	84.02	2.42
MLP	83.04	72.40	83.30	94.80	72.92	81.64	4.08
RBF	79.28	74.20	83.31	95.00	71.25	75.56	4.50
SVM	85.07	76.00	86.37	95.00	71.25	83.20	**2.17**
C4.5	85.51	73.30	84.22	94.50	68.75	82.14	3.67

the performance metrics of the methods measured across the data sets.

The Friedman test is based on the average ranked performances of a collection of techniques on each data set separately. The Friedman statistic (χ_F^2) is distributed according to the Chi-square distribution with $K-1$ degrees of freedom, when N (number of data sets) and K (number of algorithms) are big enough. The null-hypothesis being tested is that all strategies are equivalent and the observed differences are merely random. The main drawback of the Friedman and other related tests is that they only can detect significant differences over the whole set of comparisons, but they cannot compare a control technique with the $K-1$ remaining algorithms.

If the null-hypothesis of the Friedman test is rejected, we can then proceed with a post hoc test in order to find the particular pairwise comparisons that produce significant differences. For example, the Bonferroni-Dunn test can be used when all classifiers are compared with a control model (Demšar, 2006). The Bonferroni-Dunn test states that the performances of two or more algorithms are significantly different if their average ranks differ by at least the critical difference, which is given by

$$CD = q_{\alpha,\infty,K}\sqrt{\frac{K(K+1)}{6N}}$$

where the value $q_{\alpha,\infty,K}$ is based on the studentized range statistic divided by $\sqrt{2}$.

4. Results

Although the purpose of this paper is not to evaluate the performance of the base classifiers, Table 3 reports the accuracy rate and the average rank (Friedman score) of each model as a baseline for further comparisons. The technique achieving the lowest average rank (highlighted in bold face) corresponds to SVM, whereas the NBC appears as the individual classifier with the worst overall performance.

Table 4 shows the accuracy results and the Friedman ranks of the different ensembles. For each ensemble method, the base classifier with the lowest average ranking across the six credit scoring data sets is highlighted in bold face. As can be seen, the C4.5 decision tree presents the lowest rank when used with the AdaBoost, random subspace and rotation forest ensembles. The MLP neural network appears to be the best base classifier with bagging and random subspace, whereas logistic regression has the lowest rank with the DECORATE algorithm. It is worth noting that SVM, which is the individual model with the highest overall accuracy (lowest average rank), performs worse than most classifiers when used in an ensemble approach. On the other hand, the behaviour of NBC in all ensemble methods is similar to that illustrated in Table 3.

Table 4
Accuracy rate and Friedman average ranking for the classifier ensembles.

		Australian	German	Japanese	Iranian	Polish	UCSD	Average rank
AdaBoost	1-NN	79.57	65.30	78.41	93.00	73.75	77.49	6.00
	NBC	81.30	72.20	83.77	23.20	71.67	76.59	5.75
	logR	84.93	75.70	87.29	94.20	72.92	83.86	2.50
	MLP	83.04	71.50	83.61	94.80	75.42	81.56	4.08
	RBF	81.01	73.90	83.76	94.90	71.67	78.56	4.75
	SVM	84.06	76.10	84.83	95.00	72.08	83.20	2.67
	C4.5	82.90	72.30	85.91	95.10	75.42	85.83	**2.25**
Bagging	1-NN	82.03	70.80	80.85	93.00	75.00	80.04	5.58
	NBC	82.03	75.30	81.77	24.70	67.50	67.15	6.17
	logR	84.93	76.20	87.75	94.30	73.33	84.07	3.00
	MLP	86.38	75.40	85.75	94.80	78.75	83.37	**2.50**
	RBF	80.43	75.30	82.85	95.00	71.25	75.69	5.00
	SVM	85.07	76.60	86.37	95.00	72.92	82.87	2.92
	C4.5	85.94	73.70	86.83	94.70	77.08	86.28	2.83
Random subspace	1-NN	82.90	73.70	83.31	94.80	77.50	83.45	4.08
	NBC	81.16	73.40	80.39	23.60	65.83	68.62	6.67
	logR	85.94	75.50	85.45	94.80	72.92	83.78	3.08
	MLP	86.81	74.30	85.90	95.00	75.00	84.48	**2.00**
	RBF	82.61	72.90	82.70	95.00	71.25	75.40	5.33
	SVM	84.20	72.20	83.45	95.00	70.83	78.40	4.83
	C4.5	85.65	73.90	85.60	95.10	76.25	85.26	**2.00**
DECORATE	1-NN	78.99	69.80	79.48	93.00	77.50	78.89	5.83
	NBC	80.72	74.50	83.15	94.80	67.50	80.25	5.17
	logR	84.64	77.40	87.13	94.70	74.17	83.78	**2.42**
	MLP	83.19	71.50	83.60	94.70	73.75	81.56	4.58
	RBF	80.87	75.20	83.77	95.00	68.75	78.77	4.42
	SVM	85.07	76.00	86.37	95.00	71.25	83.20	2.58
	C4.5	85.94	73.20	84.23	94.20	74.58	84.89	3.00
Rotation forest	1-NN	77.83	69.80	81.01	94.50	79.17	81.11	5.33
	NBC	79.71	73.30	83.31	24.60	62.08	64.19	6.50
	logR	84.93	76.10	87.29	94.20	72.92	84.11	3.25
	MLP	85.65	74.40	84.38	95.00	73.33	85.01	3.17
	RBF	80.43	74.20	84.53	95.00	72.92	75.40	4.58
	SVM	85.22	75.30	86.37	95.00	71.67	83.00	3.50
	C4.5	85.94	74.50	87.13	95.20	76.67	86.16	**1.67**

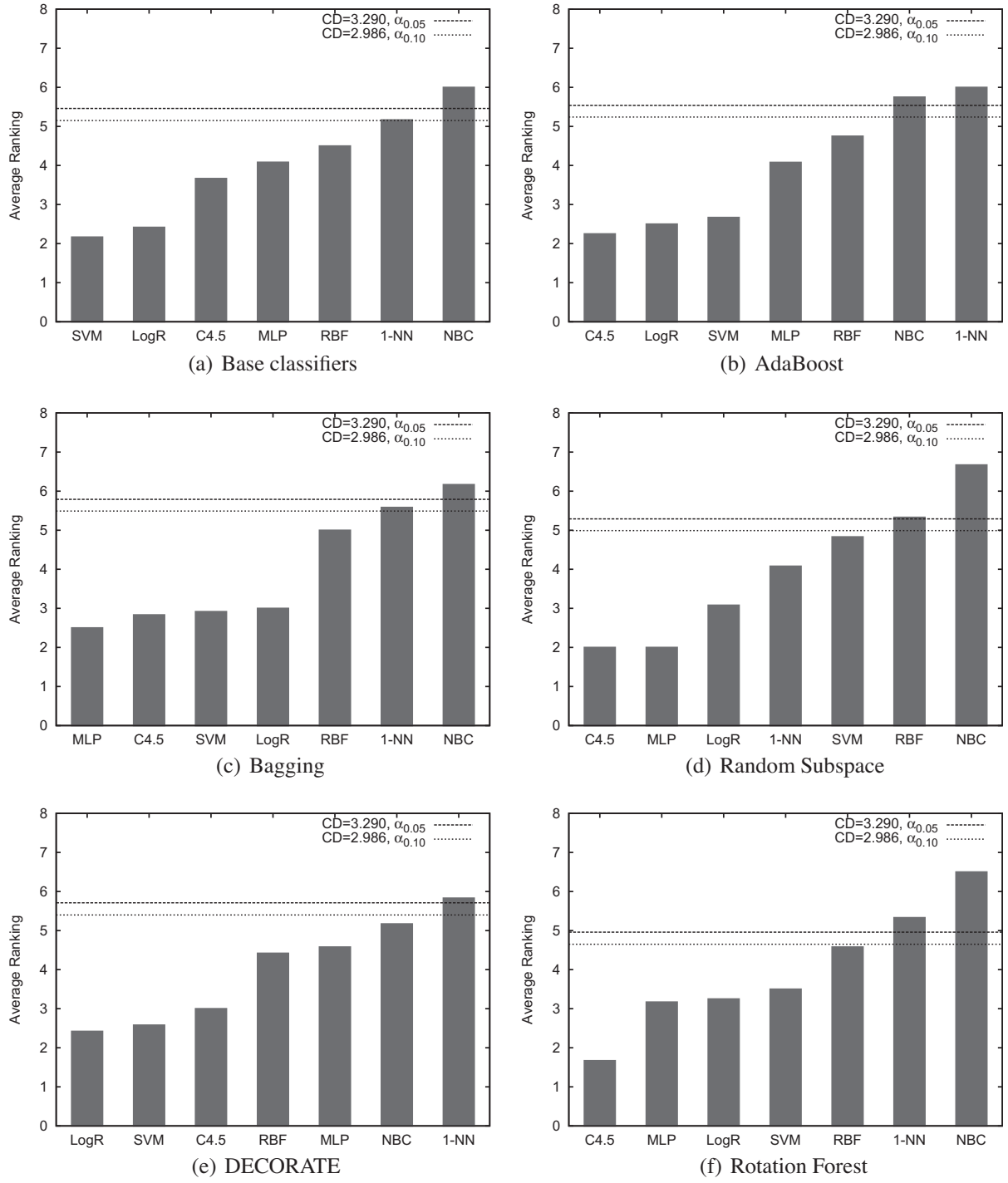

Fig. 1. Bonferroni-Dunn graphic corresponding to the base classifiers and the five ensembles.

After applying the Friedman test in order to discover whether there exist significant differences in the accuracy results, the Bonferroni-Dunn post hoc test has been employed to report any significant differences with respect to the best performing base classifier for each ensemble approach. The results of this test are then depicted to illustrate the differences among the Friedman average ranks. Fig. 1(b)–(f) plot base classifiers against average rankings, whereby all models are sorted according to their ranks. The two horizontal lines, which are at height equal to the sum of the lowest rank and the critical difference computed by the Bonferroni-Dunn test, represent the threshold for the best performing classifier at each significance level ($\alpha = 0.05$ and $\alpha = 0.10$). This means that all algorithms above these cut lines perform significantly worse than the best model. The same analysis has also been done for the individual classifiers in order to illustrate their behaviour when they are not used to construct an ensemble (see Fig. 1(a)).

From the Bonferroni-Dunn graphics plotted in Fig. 1, the following findings can be remarked:

- AdaBoost (Fig. 1(b)): the C4.5 decision tree is significantly better than 1-NN and NBC with both $\alpha = 0.05$ and $\alpha = 0.10$. The logistic regression and SVM models are also significantly better than 1-NN and NBC with $\alpha = 0.10$.

Table 5
Type-I error and Friedman average ranking for the classifier ensembles.

		Australian	German	Japanese	Iranian	Polish	UCSD	Average rank
AdaBoost	1-NN	0.17	0.54	0.20	0.68	0.31	0.42	5.08
	NBC	0.03	0.54	0.05	0.08	0.44	0.28	2.75
	logR	0.16	0.48	0.15	0.94	0.31	0.36	4.08
	MLP	0.15	0.45	0.15	0.76	0.27	0.44	3.33
	RBF	0.17	0.55	0.15	1.00	0.30	0.55	5.83
	SVM	0.16	0.50	0.15	1.00	0.25	0.43	4.17
	C4.5	0.15	0.50	0.12	0.72	0.30	0.33	2.75
Bagging	1-NN	0.15	0.55	0.19	0.70	0.29	0.40	4.50
	NBC	0.04	0.54	0.04	0.08	0.52	0.07	2.50
	logR	0.16	0.50	0.14	0.94	0.31	0.35	4.33
	MLP	0.12	0.49	0.14	0.84	0.22	0.38	2.83
	RBF	0.10	0.59	0.10	1.00	0.31	0.97	5.00
	SVM	0.21	0.51	0.20	1.00	0.22	0.43	5.17
	C4.5	0.14	0.56	0.13	0.92	0.26	0.32	3.66
Random subspace	1-NN	0.16	0.72	0.15	0.82	0.20	0.34	3.67
	NBC	0.05	0.71	0.03	0.08	0.59	0.10	2.33
	logR	0.13	0.66	0.11	1.00	0.31	0.44	4.08
	MLP	0.11	0.68	0.12	1.00	0.28	0.40	3.75
	RBF	0.12	0.79	0.11	1.00	0.36	1.00	5.25
	SVM	0.19	0.87	0.16	1.00	0.26	0.81	5.75
	C4.5	0.11	0.75	0.11	0.98	0.28	0.32	3.17
DECORATE	1-NN	0.21	0.54	0.19	0.68	0.23	0.42	3.75
	NBC	0.03	0.57	0.05	1.00	0.52	0.61	4.58
	logR	0.15	0.43	0.14	0.94	0.27	0.36	2.92
	MLP	0.15	0.45	0.16	0.76	0.30	0.40	3.58
	RBF	0.07	0.56	0.08	1.00	0.36	0.69	4.67
	SVM	0.21	0.50	0.20	1.00	0.25	0.44	4.92
	C4.5	0.12	0.57	0.15	0.82	0.29	0.34	3.58
Rotation forest	1-NN	0.21	0.59	0.18	0.80	0.20	0.37	4.25
	NBC	0.04	0.77	0.04	0.08	0.66	0.08	3.00
	logR	0.16	0.48	0.15	0.94	0.31	0.36	4.00
	MLP	0.13	0.47	0.13	1.00	0.32	0.35	3.58
	RBF	0.12	0.70	0.11	1.00	0.35	1.00	4.83
	SVM	0.21	0.56	0.20	1.00	0.24	0.47	5.33
	C4.5	0.14	0.55	0.13	0.92	0.24	0.31	3.00

- Bagging (Fig. 1(c)): the MLP neural network is significantly better than NBC with $\alpha = 0.05$ and significantly better than 1-NN and NBC with $\alpha = 0.10$. Besides, the C4.5 decision tree is significantly better than NBC with $\alpha = 0.10$.
- Random subspace (Fig. 1(d)): the MLP and C4.5 classifiers are significantly better than NBC and RBF with both $\alpha = 0.05$ and $\alpha = 0.10$. The logistic regression method is significantly better than NBC with $\alpha = 0.10$.
- DECORATE (Fig. 1(e)): logistic regression is significantly better than 1-NN with $\alpha = 0.05$ and $\alpha = 0.10$. On the other hand, SVM is significantly better than 1-NN with $\alpha = 0.10$.
- Rotation forest (Fig. 1(f)): the C4.5 decision tree is significantly better than the 1-NN and NBC models with both $\alpha = 0.05$ and $\alpha = 0.10$. The MLP, logistic regression and SVM classifiers are also significantly better than NBC with $\alpha = 0.10$.

In summary, when considering the accuracy, it should be concluded that 1-NN and NBC classifiers perform significantly the worst, regardless of the ensemble method used. It is also important to note that for most ensemble approaches, C4.5 appears to be the best base classifier, closely followed by MLP, logistic regression and SVM.

Comparing these results with the average rankings of the individual classifiers given in Table 3 and plotted in Fig. 1(a), one can observe that 1-NN and NBC correspond to the worst individual classifiers and also the worst base classifiers when used in any ensemble method. However, whilst SVM is the best individual model, the C4.5 decision tree constitutes the best general solution for combining classifiers. This allows to conclude that not always the best individual prediction method behaves the best when used in an ensemble.

Table 5 reports the type-I error and Friedman average ranking of the ensembles. Paradoxically, the NBC appears to be the best base classifier in terms of type-I error, giving the lowest average ranks. However, the Friedman ranks for C4.5 and MLP, which are usually the methods with the highest accuracy, are very close to those of NBC. According to Friedman test, there are not significant differences among the models here considered and therefore, no further statistical analysis has been carried out for this performance measure.

5. Conclusions and further extensions

The future of credit risk analysis is an increased reliance on computerized credit scoring models. Automated decision-making will never take the place of the credit manager, but it can help make quick decisions to approve or disqualify the majority of transactions that fall above or below certain credit score parameters.

The present work has focused on studying the behaviour of several well-known prediction models when used to construct classifier ensembles. With this aim, seven classification methods and five ensemble approaches have been applied to six credit scoring problems. The experimental results in terms of both accuracy and type-I error suggest that the C4.5 decision tree performs the best, although the behaviour of MLP, logistic regression and SVM is not very far that of the best algorithm. On the other hand, the 1-NN and NBC models are significantly the worst in all ensembles.

Some interesting directions for further research have emerged from this study, such as: (i) to extend the present analysis to other individual classifiers and other ensemble approaches; (ii) to explore the reasons why the best individual classifier is not the

best base classifier in an ensemble; and (iii) to compare the ensembles studied in the present work with other methods that combine different classifiers (for example, stacking or stacked generalization combines multiple base classifiers of different types on a single data set).

Acknowledgment

This work has partially been supported by the Spanish CICYT under grant TIN2009-14205.

References

Abdou, H. A., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. *Intelligent Systems in Accounting, Finance & Management, 18*(2–3), 59–88.

Abrahams, C. R., & Zhang, M. (2008). *Fair lending compliance: Intelligence and implications for credit risk management*. Hoboken, NJ: Wiley.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance, 23*(4), 589–611.

Bian, S., & Wang, W. (2007). On diversity and accuracy of homogeneous and heterogeneous ensembles. *International Journal of Hybrid Intelligent Systems, 4*(2), 103–128.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Caouette, J., Altman, E., Narayanan, P., & Nimmo, R. (2008). *Managing credit risk: The great challenge for global financial markets*. Hoboken, NJ: Wiley.

Chi, B.-W., & Hsu, C.-C. (2012). A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. *Expert Systems with Applications, 39*(3), 2650–2661.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*(1), 1–30.

Dietterich, T. G. (1997). Machine-learning research: Four current directions. *AI Magazine, 18*(4), 97–136.

Doumpos, M., & Zopounidis, C. (2007). Model combination for credit risk assessment: A stacked generalization approach. *Annals of Operations Research, 151*(1), 289–306.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. of the 13th international conference on machine learning. Bari, Italy* (pp. 148–156).

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences, 180*(10), 2044–2064.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations Newsletter, 11*(1), 10–18.

Hand, D. J. (2005). Good practice in retail credit scorecard assessment. *Journal of the Operational Research Society, 56*(9), 1109–1117.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(8), 832–844.

Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with data mining approach based on support vector machines. *Expert Systems and Applications, 33*, 847–856.

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: A market comparative study. *Decision Support Systems, 37*(4), 543–558.

Hung, C., & Chen, J.-H. (2009). A selective ensemble based on expected probabilities for bankruptcy prediction. *Expert Systems and Applications, 36*(3), 5297–5303.

Ince, H., & Aktan, B. (2009). A comparison of data mining techniques for credit scoring in banking: A managerial perspective. *Journal of Business Economics and Management, 10*(3), 233–240.

Kittler, J. (1998). Combining classifiers: A theoretical framework. *Pattern Analysis and Applications, 1*(1), 18–27.

Kuncheva, L. I. (2004). *Combining pattern classifiers: Methods and algorithms*. Hoboken, NJ: Wiley.

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning, 51*(2), 181–207.

Martens, D., van Gestel, T., de Backer, M., Haesen, R., Vanthienen, J., & Baesens, B. (2010). Credit rating prediction using ant colony optimization. *Journal of the Operational Research Society, 61*(4), 561–573.

Melville, P., & Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. *Information Fusion, 6*(1), 99–111.

Ong, C.-S., Huang, J.-J., & Tzeng, G.-H. (2005). Building credit scoring models using genetic programming. *Expert Systems with Applications, 29*(1), 41–47.

Pietruszkiewicz, W., 2008. Dynamical systems and nonlinear Kalman filtering applied in classification. In *Proc. of 7th IEEE international conference on cybernetic intelligent systems. London, UK* (pp. 263–268).

Rodriguez, J. J., Kuncheva, L. I., & Alonso, C. J. (2006). Rotation forest: A new classifier ensemble method. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 28*(10), 1619–1630.

Rosenberg, E., & Gleit, A. (1994). Quantitative methods in credit management: A survey. *Operations Research, 42*(4), 589–613.

Sabzevari, H., Soleymani, M., & Noorbakhsh, E. (2007). A comparison between statistical and data mining methods for credit scoring in case of limited available data. In *Proc. of the 3rd CRC credit scoring conference. Edinburgh, UK*.

Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). *Credit scoring and its applications*. Philadelphia, PA: SIAM.

Twala, B. (2010). Multiple classifier application to credit risk assessment. *Expert Systems with Applications, 37*(4), 3326–3336.

Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications, 38*(1), 223–230.

West, D., Dellana, S., & Qian, J. (2005). Neural network ensemble strategies for financial decision applications. *Computers and Operations Research, 32*(10), 2543–2559.

Yang, Z., Wang, Y., Bai, Y., & Zhang, X. (2004). Measuring scorecard performance. In *Proc. of 4th international conference on computational science. Krakow, Poland* (pp. 900–906).