Expert Systems with Applications 39 (2012) 10916–10922

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Two-level classifier ensembles for credit risk assessment

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:
Credit scoring
Classifier ensemble
Bagging
Boosting
Random subspace
Rotation forest

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 generally more accurate than single prediction models. The present paper goes one step beyond by introducing composite ensembles that jointly use different strategies for diversity induction. Accordingly, the combination of data resampling algorithms (bagging and AdaBoost) and attribute subset selection methods (random subspace and rotation forest) for the construction of composite ensembles is explored with the aim of improving the prediction performance. The experimental results and statistical tests show that this new two-level classifier ensemble constitutes an appropriate solution for credit scoring problems, performing better than the traditional single ensembles and very significantly better than individual classifiers.

1. Introduction

The recent world financial crisis has aroused increasing attention of banks and financial institutions on credit risk, which remains the most important and hard to manage and evaluate. The main problem comes from the difficulty to distinguish the creditworthy applicants from those who will probably default on repayments. One of the primary tools for credit risk management is credit scoring, which allows to assess credit risk, improve cash flow, reduce possible risks and make managerial decisions (Thomas, Edelman, & Crook, 2002). The decision to grant credit to an applicant was originally based upon subjective judgments made by human experts, using past experiences and some guiding principles. Common practice was to consider the classic five Cs of credit: character, capacity, capital, collateral and conditions (Rosenberg & Gleit, 1994). This method suffers, however, from high training costs, frequent incorrect decisions, and inconsistent decisions made by different experts for the same application.

Credit scoring is essentially a set of techniques that help lenders decide whether or not to grant credit to new applicants. Therefore, the objective of a credit scoring system is to distinguish "good" applicants from "bad" applicants, depending on the probability of default with their repayments (Hand & Henley, 1997). From a practical viewpoint, the process of credit scoring can be deemed as a prediction or classification problem where a new input sample (the credit applicant) must be categorized into one of the predefined classes based on a number of observed variables or attributes related to that sample. The input of the classifier consists of a variety of information that describes socio-demographic characteristics and economic conditions of the applicant, and the classifier will produce the output in terms of the applicant creditworthiness.

Because of the vast amount of information available, financial institutions have currently a need for advanced analytical tools that support the credit risk management processes in order to comply with the Basel regulatory requirements. As a consequence, many automatic credit scoring systems have been proposed in the literature. The most classical approaches are based on statistical models, such as logistic regression, linear discriminant analysis, and multivariate adaptive regression splines. However, the problem with using statistical techniques is that some assumptions, such as the multivariate normality for independent variables, are frequently violated, what makes them theoretically invalid for finite samples (Huang, Chen, Hsu, Chen, & Wu, 2004).

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

Although previous studies conclude that artificial intelligence techniques are superior to traditional statistical models, it is unlikely to find a single classifier achieving the best results on the whole application domain. Taking this into account, classifier ensembles have emerged to exploit the different behavior of

* 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.
http://dx.doi.org/10.1016/j.eswa.2012.03.033

individual classifiers and reduce prediction errors. Recent practical investigations have demonstrated that classifier ensembles generally perform better than single prediction methods in most credit scoring problems (Doumpos & Zopounidis, 2007; Twala, 2010; Wang, Hao, Ma, & Jiang, 2011; West, Dellana, & Qian, 2005).

An ensemble of classifiers is efficient only if these have a minimum of errors in common (Ali & Pazzani, 1996; Bian & Wang, 2007). In other words, the individual classifiers have to make decisions as diverse as possible. Probably, using different training sets and using different attribute subsets are the two most typical strategies to generate a diverse set of classifiers. The distinction in purpose and performance between both approaches suggests a synergistic relationship between them that is worth to be explored. The idea is that, by using them in conjunction, the diversity induced by one method can be improved with the diversity produced by the other strategy in order to construct a composite ensemble approach significantly better than any single ensemble.

The focus of this paper is therefore primarily on exploring the joint use of both diversity induction strategies for the construction of composite ensembles in the scope of credit scoring. This can be viewed as a two-level ensemble that combines two single ensembles of different nature with the aim of improving the classification performance. Another point of investigation in this paper is whether the ordering of methods matters, that is, what are the practical implications of using first a data resampling algorithm followed by an attribute selection technique or vice versa?

We investigate these questions by using two resampling-based ensembles (bagging and AdaBoost) and two attribute-based algorithms (random subspace and rotation forest) in varied sequences. The details of these ensemble approaches are presented in Section 2. Section 3 introduces the proposed methodology. Section 4 provides a description of the experiments carried out, with their results in Section 5. Finally, Section 6 remarks the main conclusions and discusses directions for further research.

2. Classifier ensembles

An ensemble of classifiers (committee of learners, mixture of experts, multiple classifier system) consists of a set of individually trained classifiers (the base classifiers) whose decisions are combined in some way, typically by weighted or unweighted voting, when classifying new examples (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, as already said, for an ensemble to achieve 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.

Various strategies have been developed to enforce diversity on the classifiers that form the ensemble. For instance, Kuncheva (2003) identified four basic approaches: (i) using different combination schemes, (ii) using different classifier models, (iii) using different attribute subsets, and (iv) using different training sets. These two last strategies constitute the most commonly used methods. In this context, two representative ensemble algorithms that use different training sets are bagging and boosting, whereas random subspace and rotation forest constitute two well-known examples of the ensemble methods that utilize different attribute subsets. In the following subsections, these popular approaches will be briefly described.

2.1. Bagging

In its standard form, the bagging (Bootstrap Aggregating) algorithm (Breiman, 1996) generates M bootstrap samples $T_1, T_2, \ldots, T_M$ randomly drawn (with replacement) from the original training set T of size n. From each bootstrap sample T_i (also of size n), a base classifier C_i is induced by the same learning algorithm. 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 (RSM) 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 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. The author of this method suggested to select about 50% of the original features.

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. 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 L 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. Then the data are 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. Constructing two-level classifier ensembles

In their most classical form, the base classifiers that comprise an ensemble correspond to simple prediction models such as neural networks, support vector machines, *k*-nearest neighbors, Bayesian classifiers and decision trees. However, the ensemble approach to credit risk assessment here proposed extends the traditional notion of multiple classifier systems by using an ensemble as base classifier of a higher-level ensemble.

In order to exploit the advantages of the two diversity induction strategies previously mentioned (i.e., using different training sets and using different attribute subsets), we here propose to construct a prediction model that integrates the resampling-based and the attribute-based methods into a unified two-level classifier ensemble. In summary, a two-level ensemble will consist of an ensemble in the first level whose base classifier is another ensemble of different nature in the second level, which in turn employs an individual classification algorithm as base classifier. For this purpose, we have two dual realizations depending on the order in which the construction techniques are applied: (i) to use bagging or AdaBoost as base classifier of the random subspace or rotation forest methods and (ii) to use one of these as base classifier of bagging or AdaBoost.

Fig. 1 shows an example of a two-level ensemble that combines bagging and the random subspace method. By employing a random subspace ensemble as base classifier of bagging, we will first generate M bootstrap replicates of the training set T. Afterwards, each bootstrap sample will be split into L subsets by randomly selecting a proportion of the original set of attributes. By this way, new observations will be classified by taking the majority vote of the ensemble C^* built from a total number of $M \times L$ classifiers $C_{1,1}, C_{1,2}, \ldots, C_{1,L}, C_{2,1}, \ldots, C_{M,L}$ trained on sets $T_{1,1}, T_{1,2}, \ldots, T_{1,L}, T_{2,1}, \ldots, T_{M,L}$.

4. Experiments

In order to test the validity and performance of the method just proposed, several experiments have been carried out. It is worth keeping in mind that the objective of this paper is twofold: (i) to explore the joint use of two different approaches to the construction of credit scoring ensembles, and (ii) to analyze the ordering in which these techniques should be applied for the best performance. These questions have been here tackled by using the four classifier ensembles outlined in Section 2: bagging (Bag), AdaBoost (AdaB), random subspace method (RSM) and rotation forest (RF).

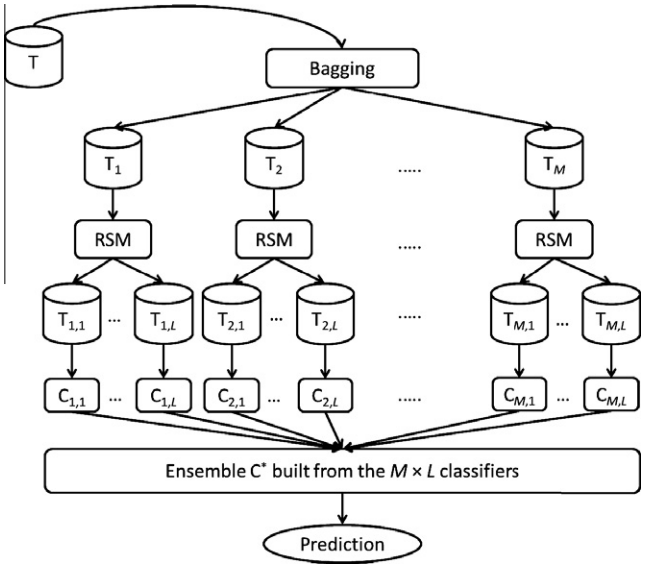

Fig. 1. A two-level ensemble consisting of a bagging ensemble in the first level and a random subspace ensemble in the second level.

Taking into account all possible combinations between the resampling strategies and the attribute-based techniques, eight different two-level ensembles can be obtained. For example, Bag (RSM) represents the approach described in Section 3, where the random subspace method acts as base classifier of a bagging ensemble.

Although decision trees have seldom been used in credit scoring applications because they are very sensitive to noise and redundant attributes in data, the C4.5 decision tree induction algorithm has been here taken as base classifier in all ensemble approaches. The reason behind this choice is that decision trees are easily interpretable by humans, they do not make any assumptions about the underlying distribution, and they can compete in performance with other techniques more widely-used in credit scoring.

Apart from the aforementioned ensembles, some individual classifiers suitable for credit scoring have also been included in this investigation in order to present a more exhaustive comparison: 1-nearest neighbor (1NN), logistic regression (logR), multilayer perceptron neural network (MLP), support vector machine (SVM) with a linear kernel, and C4.5 decision tree. In total, we have analyzed the performance of 17 prediction models for several credit scoring applications. All classifiers have been implemented using the WEKA environment (Hall et al., 2009) with the default parameter values.

4.1. 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 very large 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 2-year period (Pietruszkiewicz, 2008). Table 1 reports a summary of the main characteristics of these data sets.

4.2. 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

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

Data set	# Attributes	# Good	# Bad	% Good–% Bad
Australian	14	307	383	44.5–55.5
German	24	700	300	70.0–30.0
Japanese	15	296	357	45.3–54.7
Iranian	27	950	50	95.0–5.0
Polish	30	128	112	53.3–46.6
UCSD	38	1836	599	75.4–24.6

classification performance. The most common way around this corresponds to cross-validation (Thomas et al., 2002).

Accordingly, a fivefold 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 0.10 (Demšar, 2006).

4.3. Evaluation criteria

Standard performance evaluation criteria in the fields of credit soring 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). 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 credit scoring applications often employ the accuracy as the criterion for performance evaluation. It represents the proportion of the correctly predicted cases (good and bad) on a particular data set. However, empirical and theoretical evidences show that this measure is strongly biased with respect to data imbalance and proportions of correct and incorrect predictions (Provost & Fawcett, 1997). Because credit data are commonly imbalanced, the area under the ROC curve (AUC) has been suggested as an appropriate performance evaluator without regard to class distribution or misclassification costs (Baesens et al., 2003; Lee & Zhu, 2011). The AUC criterion for a binary problem can be defined as the arithmetic average of the mean predictions for each class (Sokolova & Lapalme, 2009):

$$\mathrm{AUC} = \frac{sensitivity + specificity}{2}$$

where $sensitivity = \frac{a}{a+b}$ measures the percentage of good applicants that have been predicted correctly, whereas $specificity = \frac{d}{c+d}$ corresponds to the percentage of bad applicants predicted as bad.

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

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

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.

Table 2
Confusion matrix for a credit scoring problem.

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

4.4. 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).

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 and are independent of any evaluation measure. In this work, we have adopted the Friedman test to compare the performance 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 χ^2 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 studentised range statistic divided by $\sqrt{2}$.

5. Results

Table 3 shows the AUC values and the Friedman ranks of the different prediction models. As can be seen, the Bag (RF) and RF (Bag) ensemble approaches correspond to the techniques with the lowest average rankings (highest AUC values), followed by RF (AdaB) and Bag (RSM). It is also worth noting that the two-level ensembles perform better than the single ensembles, except for both implementations where the AdaBoost algorithm is first applied. On the other hand, as expected, the individual classifiers achieve the lowest AUC values, being 1NN and logistic regression the worst and the best methods, respectively.

Although differences in AUC may appear to be relatively low, it should be noted that even a small increase in prediction performance can yield substantial cost savings for financial institutions. It seems therefore to be of sufficient interest the use of the two-level classifier ensembles in credit scoring applications. The highest differences are observed for the Iranian credit data set, which

Table 3
AUC values (with standard deviations) and average rankings for the classifiers.

	Australian	German	Japanese	Iranian	Polish	UCSD	Rank
1NN	0.81 (0.04)	0.64 (0.02)	0.79 (0.03)	0.64 (0.08)	0.75 (0.04)	0.73 (0.03)	16.33
log*R*	0.90 (0.03)	0.79 (0.03)	0.93 (0.02)	0.73 (0.08)	0.79 (0.05)	0.88 (0.01)	10.33
MLP	0.88 (0.02)	0.74 (0.04)	0.91 (0.03)	0.73 (0.11)	0.82 (0.04)	0.85 (0.02)	13.00
SVM	0.86 (0.02)	0.69 (0.02)	0.87 (0.02)	0.50 (0.00)	0.71 (0.06)	0.74 (0.02)	16.00
C4.5	0.89 (0.02)	0.69 (0.04)	0.86 (0.03)	0.61 (0.15)	0.71 (0.10)	0.76 (0.03)	15.50
AdaB	0.90 (0.02)	0.73 (0.03)	0.92 (0.02)	0.74 (0.06)	0.82 (0.05)	0.90 (0.02)	11.58
Bag	0.93 (0.01)	0.74 (0.03)	0.93 (0.01)	0.77 (0.08)	0.84 (0.07)	0.90 (0.01)	8.25
RSM	0.91 (0.02)	0.76 (0.02)	0.91 (0.03)	0.72 (0.04)	0.82 (0.08)	0.90 (0.00)	11.33
RF	0.93 (0.02)	0.77 (0.03)	0.93 (0.02)	0.77 (0.14)	0.84 (0.05)	0.91 (0.01)	7.00
AdaB (RSM)	0.90 (0.01)	0.73 (0.05)	0.92 (0.02)	0.82 (0.08)	0.84 (0.07)	0.91 (0.01)	9.75
AdaB (RF)	0.91 (0.02)	0.76 (0.05)	0.93 (0.03)	0.75 (0.13)	0.86 (0.05)	0.91 (0.01)	7.50
Bag (RSM)	0.93 (0.01)	0.79 (0.03)	0.94 (0.02)	0.82 (0.06)	0.85 (0.05)	0.92 (0.01)	4.33
Bag (RF)	0.93 (0.01)	0.80 (0.03)	0.94 (0.02)	0.86 (0.06)	0.86 (0.05)	0.92 (0.01)	2.58
RSM (AdaB)	0.91 (0.02)	0.76 (0.04)	0.91 (0.04)	0.84 (0.08)	0.86 (0.06)	0.93 (0.01)	6.50
RSM (Bag)	0.93 (0.01)	0.79 (0.04)	0.92 (0.04)	0.84 (0.07)	0.84 (0.07)	0.92 (0.01)	5.75
RF (AdaB)	0.92 (0.02)	0.79 (0.03)	0.94 (0.02)	0.85 (0.04)	0.85 (0.04)	0.92 (0.01)	4.33
RF (Bag)	0.93 (0.02)	0.79 (0.03)	0.94 (0.02)	0.87 (0.04)	0.86 (0.05)	0.92 (0.01)	2.92

corresponds to a strongly imbalanced problem (95% of good applicants with only 5% of bad applicants); for example, the two-level RF (Bag) method (the model with the highest AUC for this database) performs better than bagging and rotation forest by 0.10, better than Adaboost by 0.13 and better than RSM by 0.15. If we compare the RF (Bag) model with the individual classifiers, the differences are even much more significant: 0.37, 0.26, 0.23 and 0.14 with respect to SVM, C4.5, 1NN and log*R*, respectively.

After applying the Friedman test in order to discover whether there exist significant differences in the AUC results, the Bonferroni–Dunn post-hoc test has been employed to report any significant differences with respect to the best performing method (bagging with rotation forest) for each prediction model. The results of this test have been then depicted to illustrate the differences among the Friedman average ranks. Fig. 2 plots techniques 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 method at each significance level ($\alpha = 0.05$ and $\alpha = 0.10$). This indicates that all algorithms above these cut lines perform significantly worse than the best model.

From the Bonferroni–Dunn graphic plotted in Fig. 2, one can observe that the only single ensembles not significantly worse than the best Bag (RF) model correspond to the single bagging and rotation forest algorithms. It is also interesting to remark that

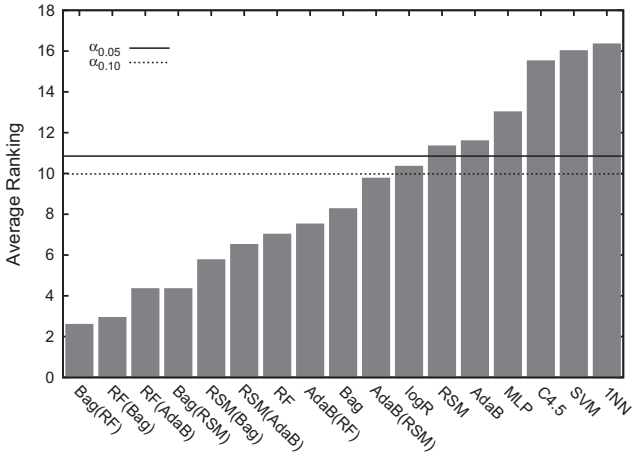

Fig. 2. Significance diagram for the Bonferroni–Dunn test with $\alpha = 0.05$ and 0.10.

the differences between both composite ensembles with bagging and rotation forest are very small. In fact, they achieved the same AUC values in 4 out of 6 databases, Bag (RF) performed better than RF (Bag) in the case of the German data set, and RF (Bag) was better than Bag (RF) for the Iranian data. Similarly, the order in which bagging and RSM are combined does not lead to substantial differences. However, it seems that it is preferable to firstly apply an attribute-based method and then the AdaBoost algorithm rather than the inverse order: both RSM (AdaB) and RF (AdaB) have been better than AdaB (RSM) and AdaB (RF) in 5 out of 6 cases.

As already commented in Section 4.3, it is useful to evaluate the performance on each individual class because the misclassification costs associated with each error type are usually different. From a theoretical point of view, it is better to utilize prediction models with lower type-I errors (percentage of bad credit applicants who have been predicted as good), but in practice it is also of great importance for the financial institutions to achieve an appropriate balance between both error types so as not to lose potentially good customers. Accordingly, Tables 4 and 5 report the type-I and type-II errors for the two-level classifier ensembles, respectively.

Analysing the results in Table 4, one can observe that both Bag (RSM) and RF (AdaB) achieved the lowest type-I errors in 2 out of 6 databases, although the differences were not statistically significant. In the case of the type-II errors given in Table 5, the best two-level ensemble seems to correspond to Bag (RF) obtaining the lowest error rates in 3 out of 6 databases, followed by RF (Bag) and Bag (RSM) with the lowest type-II errors in 2 out of 6 databases.

In summary, taking into account the three performance measures here used, the following findings can be remarked:

- The best overall methods correspond to the two-level classifier ensembles that use bagging and rotation forest, independently of the order in which they are applied.
- In general, the differences in type-I and type-II errors appear not to be statistically significant.
- When implementing the AdaBoost algorithm in a two-level ensemble, it seems that the ordering of methods matters: it is better to use first an attribute subset selection strategy (RSM or rotation forest) followed by the AdaBoost algorithm. Paradoxically, however, for the strongly imbalanced Iranian data set, AdaB (RSM) and AdaB (RF) lead to the lowest type-I errors with the same type-II error rates than the remaining approaches.

Table 4
Type-I error rates and standard deviations for the two-level ensembles.

	Australian	German	Japanese	Iranian	Polish	UCSD
AdaB (RSM)	0.12 (0.05)	0.57 (0.07)	0.11 (0.03)	0.72 (0.08)	0.28 (0.06)	0.31 (0.03)
AdaB (RF)	0.13 (0.03)	0.53 (0.09)	0.12 (0.04)	0.78 (0.08)	0.22 (0.11)	0.34 (0.06)
Bag (RSM)	0.10 (0.04)	0.71 (0.05)	0.10 (0.02)	0.96 (0.05)	0.24 (0.11)	0.30 (0.06)
Bag (RF)	0.13 (0.04)	0.55 (0.05)	0.13 (0.02)	0.94 (0.09)	0.22 (0.06)	0.31 (0.05)
RSM (AdaB)	0.11 (0.03)	0.66 (0.05)	0.11 (0.03)	0.84 (0.05)	0.23 (0.10)	0.31 (0.04)
RSM (Bag)	0.11 (0.04)	0.73 (0.04)	0.11 (0.03)	0.96 (0.05)	0.22 (0.10)	0.31 (0.07)
RF (AdaB)	0.12 (0.04)	0.57 (0.04)	0.12 (0.02)	0.82 (0.08)	0.20 (0.07)	0.07 (0.01)
RF (Bag)	0.14 (0.04)	0.57 (0.07)	0.13 (0.02)	0.94 (0.05)	0.23 (0.06)	0.31 (0.03)

Table 5
Type-II error rates and standard deviations for the two-level ensembles.

	Australian	German	Japanese	Iranian	Polish	UCSD
AdaB (RSM)	0.20 (0.05)	0.17 (0.03)	0.16 (0.05)	0.01 (0.01)	0.20 (0.08)	0.08 (0.01)
AdaB (RF)	0.16 (0.02)	0.15 (0.05)	0.16 (0.04)	0.01 (0.00)	0.22 (0.10)	0.07 (0.00)
Bag (RSM)	0.19 (0.04)	0.04 (0.02)	0.16 (0.05)	0.00 (0.00)	0.19 (0.06)	0.08 (0.01)
Bag (RF)	0.13 (0.04)	0.09 (0.05)	0.13 (0.05)	0.00 (0.00)	0.21 (0.04)	0.06 (0.00)
RSM (AdaB)	0.17 (0.05)	0.09 (0.03)	0.17 (0.07)	0.01 (0.00)	0.18 (0.09)	0.07 (0.01)
RSM (Bag)	0.16 (0.04)	0.04 (0.02)	0.18 (0.08)	0.00 (0.00)	0.19 (0.07)	0.08 (0.01)
RF (AdaB)	0.14 (0.05)	0.10 (0.02)	0.13 (0.04)	0.01 (0.00)	0.22 (0.05)	0.33 (0.05)
RF (Bag)	0.14 (0.03)	0.11 (0.03)	0.12 (0.04)	0.00 (0.00)	0.19 (0.06)	0.07 (0.01)

6. Conclusions and further extensions

In this work, a new methodology for credit assessment has been developed by combining different classifier ensemble methods, with the aim of obtaining better performance results than the single ensembles. This can be viewed as a two-level ensemble approach that combines data resampling and attribute subset selection strategies for the construction of composite ensembles. In particular, AdaBoost and bagging have been taken as representatives of data resampling algorithms for diversity induction, whereas the random subspace method and rotation forest have been used as examples of the attribute subset selection techniques.

Some interesting conclusions can be drawn from the experiments carried out. In general, the two-level classifier ensembles have produced the best results in terms of AUC, what may lead to significant cost savings in credit scoring applications. Since the choice of the particular two-level ensemble model is important, it seems that the jointly use of bagging and rotation forest in any order performs better than the other combinations. A final indication from the experiments is that using the AdaBoost algorithm before some attribute subset selection method is clearly worse than the inverse order, especially in the case of combining with rotation forest.

Several directions for further research have emerged from this study: (i) To extend the present analysis to other ensemble approaches; (ii) 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); and (iii) To explore the possibility of using multi-level classifier ensembles, that is, to extend the number of ensembles that are jointly used.

Acknowledgements

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.

Ali, K. M., & Pazzani, M. J. (1996). Error reduction through learning multiple descriptions. *Machine Learning, 24*(3), 173–202.

Baesens, B., van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*(6), 627–635.

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).

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.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society: Series A, 160*(3), 523–541.

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.

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.

Kuncheva, L. I. (2003). Combining classifiers: Soft computing solutions. In S. K. Pal & A. Pal (Eds.), *Pattern recognition: From classical to modern approaches* (pp. 427–449). World Scientific.

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

Lee, J.-S., & Zhu, D. (2011). When costs are unequal and unknown: A subtree grafting approach for unbalanced data classification. *Decision Sciences, 42*(4), 803–829.

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.

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 seventh IEEE international conference on cybernetic intelligent systems*, London, UK (pp. 263–268).

Provost, F., Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: *Proc. of the third international conference on knowledge discovery and data mining*, Newport Beach, CA (pp. 43–48).

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 third CRC credit scoring conference*, Edinburgh, UK.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management, 45*(4), 427–437.

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.