Expert Systems with Applications 39 (2012) 3446–3453

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

An experimental comparison of classification algorithms for imbalanced credit scoring data sets

Iain Brown [*], Christophe Mues

School of Management, University of Southampton, Highfield, Southampton SO17 1BJ, UK

ARTICLE INFO

Keywords:
Credit scoring
Imbalanced datasets
Classification
Benchmarking

ABSTRACT

In this paper, we set out to compare several techniques that can be used in the analysis of imbalanced credit scoring data sets. In a credit scoring context, imbalanced data sets frequently occur as the number of defaulting loans in a portfolio is usually much lower than the number of observations that do not default. As well as using traditional classification techniques such as logistic regression, neural networks and decision trees, this paper will also explore the suitability of gradient boosting, least square support vector machines and random forests for loan default prediction.

Five real-world credit scoring data sets are used to build classifiers and test their performance. In our experiments, we progressively increase class imbalance in each of these data sets by randomly undersampling the minority class of defaulters, so as to identify to what extent the predictive power of the respective techniques is adversely affected. The performance criterion chosen to measure this effect is the area under the receiver operating characteristic curve (AUC); Friedman's statistic and Nemenyi post hoc tests are used to test for significance of AUC differences between techniques.

The results from this empirical study indicate that the random forest and gradient boosting classifiers perform very well in a credit scoring context and are able to cope comparatively well with pronounced class imbalances in these data sets. We also found that, when faced with a large class imbalance, the C4.5 decision tree algorithm, quadratic discriminant analysis and *k*-nearest neighbours perform significantly worse than the best performing classifiers.

1. Introduction

The aim of credit scoring is essentially to classify loan applicants into two classes, i.e., good payers (i.e., those who are likely to keep up with their repayments) and bad payers (i.e., those who are likely to default on their loans). In the current financial climate, and with the recent introduction of the Basel II Accord, financial institutions have even more incentives to select and implement the most appropriate credit scoring techniques for their credit portfolios. It is stated in Henley and Hand (1997) that companies could make significant future savings if an improvement of only a fraction of a percent could be made in the accuracy of the credit scoring techniques implemented. However, in the research literature, portfolios that can be considered as very low risk, or low default portfolios (LDPs), have had relatively little attention paid to them in particular with regards to which techniques are most appropriate for scoring them. The underlying problem with LDPs is that they contain a much smaller number of observations in the class of defaulters than in that of the good payers. A large class imbalance is therefore present which some techniques may not be able to successfully handle. Typical examples of low default portfolios include high-quality corporate borrowers, banks, sovereigns and some categories of specialised lending (Van Der Burgt, 2007) but in some countries even certain retail lending portfolios could turn out to have very low numbers of defaults compared to the majority class. In a recent FSA publication regarding conservative estimation of low default portfolios, regulatory concerns were raised about whether firms can adequately asses the risk of LDPs (Benjamin, Cathcart, & Ryan, 2006).

A wide range of classification techniques have already been proposed in the credit scoring literature, including statistical techniques, such as linear discriminant analysis and logistic regression, and non-parametric models, such as *k*-nearest neighbour and decision trees. But it is currently unclear from the literature which technique is the most appropriate for improving discrimination for LDPs. Table 1 provides a selection of techniques currently applied in a credit scoring context, along with references showing some of their reported applications in the literature.

Hence, the aim of this paper is to conduct a study of various classification techniques based on five real-life credit scoring data sets. These data sets will then have the size of their minority class

[*] Corresponding author. Address: 44 Holters Mill, The Spires, Canterbury, Kent CT2 8SP, UK. Tel.: +44 (0) 7840057162.

E-mail addresses: i.brown@soton.ac.uk (I. Brown), C.Mues@soton.ac.uk (C. Mues).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.033

Table 1
Credit scoring techniques and their applications.

Classification techniques	Application in a credit scoring context
Logistic regression (LOG)	Arminger, Enache, and Bonne (1997), Baesens et al. (2003), Desai et al. (1996), Steenackers and Goovaerts (1989), West (2000), Wiginton (1980)
Decision trees (C4.5, CART, etc.)	Arminger et al. (1997), Baesens et al. (2003), West (2000), Yobas et al. (2000)
Neural networks (NN)	Altman (1994), Arminger et al. (1997), Baesens et al. (2003), Desai et al. (1996), West (2000), Yobas et al. (2000)
Linear discriminant analysis (LDA)	Altman (1968), Baesens et al. (2003), Desai et al. (1996), West (2000), Yobas et al. (2000)
Quadratic discriminant analysis (QDA)	Altman (1968), Baesens et al. (2003)
k-Nearest neighbours (*k*-NN)	Baesens et al. (2003), Chatterjee and Barcun (1970), West (2000)
Support vector machines (SVM, LS-SVM, etc.)	Baesens et al. (2003), Yang (2007)

of defaulters further reduced by decrements of 5% (from an original 70/30 good/bad split) to see how the performance of the various classification techniques is affected by increasing class imbalance.

The five real-life credit scoring data sets used in this empirical research study include two data sets from Benelux (Belgium, Netherlands and Luxembourg) institutions, the German Credit and Australian Credit data sets which are publicly available at the UCI repository (http://kdd.ics.uci.edu/), and the fifth data set is a behavioural scoring data set, which was also obtained from a Benelux institution.

The techniques that will be applied in this paper are logistic regression (LOG), linear and quadratic discriminant analysis (LDA, QDA), least square support vector machines (LS-SVM), decision trees (C4.5), neural networks (NN), nearest-neighbour classifiers (*k*-NN10, *k*-NN100), a gradient boosting algorithm and random forests. We are especially interested in the power and usefulness of the gradient boosting and random forest classifiers which have yet to be thoroughly investigated in a credit scoring context.

All techniques will be evaluated in terms of their area under the receiver operating characteristic curve (AUC). This is a measure of the discrimination power of a classifier without regard to class distribution or misclassification cost (Baesens et al., 2003).

To make statistical inferences from the observed difference in AUC, we followed the recommendations given in a recent article (Demšar, 2006) that looked at the problem of benchmarking classifiers on multiple data sets. The recommendations given were for a set of simple robust non-parametric tests for the statistical comparison of the classifiers (Demšar, 2006). The AUC measures will therefore be compared using Friedman's average rank test, and Nemenyi's post hoc test will be employed to test the significance of the differences in rank between individual classifiers. Finally, a variant of Demšar's significance diagrams will be plotted to visualise their results.

The organisation of this paper is as follows. Section 2 will begin by providing a literature review of the work that has been conducted on the topic of classification for imbalanced data sets. A brief explanation will then be given for the ten classification techniques to be used in the analysis of the data sets. Secondly, the empirical set up and criteria used for comparing the classification performance will be described. Thirdly, the results of our experiments are presented and discussed. Finally, conclusions will be drawn from the study and recommendations for further research work will be outlined.

2. Literature review

A wide range of different classification techniques for scoring credit data sets has been proposed in the literature, a non-exhaustive list of which was provided earlier in Table 1. In addition, some benchmarking studies have been undertaken to empirically compare the performance of these various techniques (e.g., Baesens et al., 2003), but they did not focus specifically on how these techniques compare on heavily imbalanced samples, or to what extent any such comparison is affected by the issue of class imbalance. For example, in Baesens et al. (2003) seventeen techniques including both well-known techniques such as logistic regression and discriminant analysis and more advanced techniques such as least square support vector machines were compared on eight real-life credit scoring data sets. Although more complicated techniques such as radial basis function least square support vector machines (RBF LS-SVM) and neural networks (NN) yielded good performances in terms of AUC, simpler linear classifiers such as linear discriminant analysis (LDA) and logistic regression (LOG) also gave very good performances. However, there are often conflicting opinions when comparing the conclusions of studies promoting differing techniques. For example, in Yobas, Crook, and Ross (2000), the authors found that linear discriminant analysis (LDA) outperformed neural networks in the prediction of loan default, whereas in Desai, Crook, and Overstreet (1996), neural networks were reported to actually perform significantly better than LDA. Furthermore, many empirical studies only evaluate a small number of classification techniques on a single credit scoring data set. The data sets used in these empirical studies are also often far smaller and less imbalanced than those data sets used in practice. Hence, the issue of which classification technique to use for credit scoring, particularly with a small number of bad observations, remains a challenging problem (Baesens et al., 2003).

The topic of which good/bad distribution is the most appropriate in classifying a data set has been discussed in some detail in the machine learning and data mining literature. In Weiss and Provost (2003) it was found that the naturally occurring class distributions in the 25 data sets looked at, often did not produce the best-performing classifiers. More specifically, based on the AUC measure (which was preferred over the use of the error rate), it was shown that the optimal class distribution should contain between 50% and 90% minority class examples within the training set. Alternatively, a progressive adaptive sampling strategy for selecting the optimal class distribution is proposed in Provost, Jensen, and Oates (1999). Whilst this method of class adjustment can be very effective for large data sets, with adequate observations in the minority class of defaulters, in some low default portfolios there are only a very small number of loan defaults to begin with.

Various kinds of techniques have been compared in the literature to try and ascertain the most effective way of overcoming a large class imbalance. Chawla, Bowyer, Hall, and Kegelmeyer (2002) proposed a synthetic minority over-sampling technique (SMOTE) which was applied to example data sets in fraud, telecommunications management, and detection of oil spills in satellite images. In Japkowicz (2000), over-sampling and downsizing were compared to the author's own method of "learning by recognition" in order to determine the most effective technique. The findings, however, were inconclusive but demonstrated that both over-sampling the minority class and downsizing the majority class can be very effective. Subsequently, Batista (2004) identified ten alternative techniques in dealing with class imbalances and trialed them on thirteen data sets. The techniques chosen included a variety of under-sampling and over-sampling methods. Findings suggested that generally over-sampling methods provide more accurate results than under-sampling methods. Also, a combination of either SMOTE (Chawla et al., 2002) and Tomek links or SMOTE and ENN (a nearest-neighbour cleaning rule), were proposed.

3. Overview of classification techniques

This study aims to compare the performance of a wide range of classification techniques within a credit scoring context, thereby assessing to what extent they are affected by increasing class imbalance. For the purpose of this study, ten classifiers have been selected which provide a balance between well-established credit scoring techniques such as logistic regression, decision trees and neural networks, and newly developed machine learning techniques such as least square support vector machines, gradient boosting and random forests. A brief explanation of each of the techniques applied in this paper is presented below.

3.1. Logistic regression

For this paper, we will be focusing on the binary response of whether a creditor turns out to be a good or bad payer (i.e., non-defaulter vs. defaulter). For this binary response model, the response variable, y, can take on one of two possible values; i.e., $y = 0$ if the customer is a bad payer, $y = 1$ if he/she is a good payer. Let us assume $\mathbf{x}$ is a column vector of M explanatory variables and $\pi = Pr(y = 1|\mathbf{x})$ is the response probability to be modelled. The number of observations is denoted by N. The logistic regression model then takes the form:

$$\operatorname{logit}(\pi) \equiv \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta^T \mathbf{x}, \tag{1}$$

where α is the intercept parameter and β^T contains the variable coefficients (Hosmer & Stanley, 2000).

3.2. Linear and quadratic discriminant analysis

Discriminant analysis assigns an observation to the response, $y(y \in \{0,1\})$, with the largest posterior probability; i.e., classify into class 0 if $p(0|\mathbf{x}) > p(1|\mathbf{x})$, or class 1 if the reverse is true. According to Bayes' theorem, these posterior probabilities are given by

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}. \tag{2}$$

Assuming now that the class-conditional distributions $p(\mathbf{x}|y = 0)$, $p(\mathbf{x}|y = 1)$ are multivariate normal distributions with mean vector μ_0, μ_1, and covariance matrix Σ_0, Σ_1, respectively, the classification rule becomes: classify as $y = 0$ if the following is satisfied:

$$\begin{aligned}(\mathbf{x}-\mu_0)^T \sum_0^{-1} (\mathbf{x}-\mu_0) - (\mathbf{x}-\mu_1)^T \sum_1^{-1} (\mathbf{x}-\mu_1) \\ < 2(\log(P(y=0) - \log(P(y=1)))) + \log|\Sigma_1| - \log|\Sigma_0|\end{aligned} \tag{3}$$

Linear discriminant analysis is then obtained if the simplifying assumption is made that both covariance matrices are equal, i.e., $\Sigma_0 = \Sigma_1 = \Sigma$, which has the effect of cancelling out the quadratic terms in the expression above.

3.3. Neural networks (Multi-layer perceptron)

Neural networks (NN) are mathematical representations modelled on the functionality of the human brain (Bishop, 1995). The added benefit of a NN is its flexibility in modelling virtually any non-linear association between input variables and target variable. Although various architectures have been proposed, our study focuses on probably the most widely used type of NN, i.e., the multilayer perceptron (MLP). A MLP is typically composed of an input layer (consisting of neurons for all input variables), a hidden layer (consisting of any number of hidden neurons), and an output layer (in our case, one neuron). Each neuron processes its inputs and transmits its output value to the neurons in the subsequent layer. Each such connection between neurons is assigned a weight during training. The output of hidden neuron i is computed by applying an activation function $f^{(1)}$ (for example the logistic function) to the weighted inputs and its bias term $b_i^{(1)}$:

$$h_i = f^{(1)}\left(b_i^{(1)} + \sum_{j=1}^{M} \mathbf{W}_{ij} x_j\right), \tag{4}$$

where $\mathbf{W}$ represents a weight matrix in which $\mathbf{W}_{ij}$ denotes the weight connecting input j to hidden neuron i. For the analysis conducted in this paper, a binary prediction will be made; hence, for the activation function in the output layer, we will be using the logistic (sigmoid) activation function, $f^{(2)}(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{x}}}$ to obtain a response probability:

$$\pi = f^{(2)}\left(b^{(2)} + \sum_{j=1}^{n_h} \mathbf{v}_j h_j\right), \tag{5}$$

with n_h the number of hidden neurons and $\mathbf{v}$ the weight vector where $\mathbf{v}_j$ represents the weight connecting hidden neuron j to the output neuron. During model estimation, the weights of the network are first randomly initialised and then iteratively adjusted so as to minimise an objective function, e.g., the sum of squared errors (possibly accompanied by a regularisation term to prevent over-fitting). This iterative procedure can be based on simple gradient descent learning or more sophisticated optimisation methods such as Levenberg–Marquardt or Quasi-Newton. The number of hidden neurons can be determined through a grid search based on validation set performance.

3.4. Least square support vector machines (LS-SVMs)

Support vector machines (SVMs) are a set of powerful supervised learning techniques used for classification and regression. Their basic principle is to construct a maximum-margin separating hyperplane in some transformed feature space. Rather than requiring one to specify the exact transformation though, they use the principle of kernel substitution to turn them into a general (non-linear) model. The least square support vector machine (LS-SVM) proposed by Suykens, Van Gestel, De Brabanter, De Moor, and Vandewalle (2002) is a further adaptation of Vapnik's original SVM formulation which leads to solving linear KKT (Karush–Kuhn–Tucker) systems (rather than a more complex quadratic programing problem). The optimisation problem for the LS-SVM is defined as:

$$\min_{\mathbf{w},b,\mathbf{e}} J(\mathbf{w},b,\mathbf{e}) = \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} + \gamma\frac{1}{2}\sum_{i=1}^{N} e_i^2, \tag{6}$$

subject to the following equality constraints:

$$y_i\left[\mathbf{w}^{\mathbf{T}}\varphi(\mathbf{x}_i) + b\right] = 1 - e_i, \quad i = 1,\ldots,N, \tag{7}$$

Where $\mathbf{w}$ is the weight vector in primal space, γ is the regularisation parameter, and $y_i = +1$ or -1 for good (bad) payers, respectively (Suykens et al., 2002). A solution can then be obtained after constructing the Lagrangian, and choosing a particular kernel function $K(\mathbf{x},\mathbf{x}_i)$ that computes inner products in the transformed space, based on which a classifier of the following form is obtained:

$$y(\mathbf{x}) = \text{sign}\left[\sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x},\mathbf{x}_i) + b\right],$$

where by $K(\mathbf{x},\mathbf{x}_i) = \varphi(\mathbf{x})^T\varphi(\mathbf{x}_i)$ is taken to be a positive definite kernel satisfying the Mercer theorem.The hyper parameter γ for the LS-SVM classification technique is tuned using 10-fold cross validation.

3.5. C4.5. decision trees

A decision tree consists of internal nodes that specify tests on individual input variables or attributes that split the data into smaller subsets, and a series of leaf nodes assigning a class to each of the observations in the resulting segments. For our study, we chose the popular decision tree classifier C4.5, which builds decision trees using the concept of information entropy (Quinlan, 1993). The entropy of a sample S of classified observations is given by

$$\text{Entropy } (S) = -p_1\log_2(p_1) - p_0\log_2(p_0), \tag{8}$$

where $p_1(p_0)$ are the proportions of the class values 1(0) in the sample S, respectively. C4.5 examines the normalised information gain (entropy difference) that results from choosing an attribute for splitting the data. The attribute with the highest normalised information gain is the one used to make the decision. The algorithm then recurs on the smaller subsets.

3.6. k-NN (memory based reasoning)

The k-nearest neighbours algorithm (k-NN) classifies a data point by taking a majority vote of its k most similar data points (Hastie, Tibshirani, & Friedman, 2001). The similarity measure used in this paper is the Euclidean distance between the two points:

$$d(\mathbf{x}_i,\mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \left[(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)\right]^{1/2}. \tag{9}$$

3.7. Random forests

Random forests are defined as a group of un-pruned classification or regression trees, trained on bootstrap samples of the training data using random feature selection in the process of tree generation. After a large number of trees have been generated, each tree votes for the most popular class. These tree voting procedures are collectively defined as random forests. A more detailed explanation of how to train a random forest can be found in Breiman (2001). For the Random Forests classification technique two parameters require tuning. These are the number of trees and the number of attributes used to grow each tree.

3.8. Gradient boosting

Gradient boosting (Friedman, 2001, 2002) is an ensemble algorithm that improves the accuracy of a predictive function through incremental minimisation of the error term. After the initial base learner (most commonly a tree) is grown, each tree in the series is fit to the so-called “pseudo residuals” of the prediction from the earlier trees with the purpose of reducing the error. This leads to the following model:

$$F(\mathbf{x}) = G_0 + \beta_1 T_1(\mathbf{x}) + \beta_2 T_2(\mathbf{x}) + \cdots + \beta_n T_n(\mathbf{x}), \tag{10}$$

where G_0 equals the first value for the series, $T_1,\ldots,T_n$ are the trees fitted to the pseudo-residuals, and β_i are coefficients for the respective tree nodes computed by the gradient boosting algorithm. A more detailed explanation of gradient boosting can be found in Friedman (2001, 2002). The gradient boosting classifier requires tuning of the number of iterations and the maximum branch size used in the splitting rule.

4. Experimental set-up and data sets

4.1. Data set characteristics

The characteristics of the data sets used in evaluating the performance of the aforementioned classification techniques are given below in Table 2. The Bene1 and Bene2 data sets were obtained from two major financial institutions in the Benelux region. For these two data sets, a bad customer was defined as someone who had missed three consecutive months of payments. The German credit data set and the Australian Credit data set are publicly available at the UCI repository (http://www.kdd.ics.uci.edu/). The Behav data set was also acquired from a Benelux institution. As all the data sets used have a reasonable number of observations they will each be split into a training (two thirds) and a test set (one third). This test set will remain unchanged throughout the analysis of the techniques.

4.2. Re-sampling setup and performance metrics

In order for the percentage reduction in the bad observations, in each data set, to be relatively compared, the Bene1 set, Australian credit and the Behavioural Scoring set have first been altered to give a 70/30 class distribution. This was done by either under-sampling the bad observations (from a total of 1041 bad observations in the Bene1 data set, only 892 observations have been used; and from a total of 307 bad observations in the Australian credit data set, only 164 observations have been used) or under-sampling the good observations in the behavioural scoring data set, (from a total of 1436 good observations, only 838 observations have been used).

For this empirical study, the class of defaulters in each of the training data sets was artificially reduced, by a factor of 5% up to 95% then by 2.5% and 1%, so as to create a larger difference in class distribution. As a result of this reduction, eight data sets were created for each of the five original data sets. The percentage splits created were 75%, 80%, 85%, 90%, 95%, 97.5%, 99% good observations. For this empirical study our focus is on the performance of classification techniques on data sets with a large class imbalance. Therefore detailed results will only be presented for the data set with the original 70/30 split, as a benchmark, and data sets with 85%, 90% and 99% splits. By doing so, it is possible to identify whether techniques are adversely affected in the prediction of the target variable when there is a substantially lower number of observations in one of the classes. The performance criterion chosen to measure this effect is the area under the receiver operator characteristic curve (AUC) statistic as proposed by Baesens et al. (2003).

Table 2
Characteristics of credit scoring data sets.

	Inputs	Data set size	Training set size	Test set size	Goods/ bads
Bene1	27	2974	1984	990	70/30*
Bene2	27	7190	4795	2395	70/30
Austr	14	547	366	181	70/30*
Behav	60	1197	799	398	70/30*
Germ	20	1000	668	332	70/30

* Altered data set class distribution, Bene1 original distribution was 66.6% good observations, 33.3% bad observations, Austr original distribution was 55.5% good observations, 44.5% bad observations and the Behav original distribution was 80% good observations, 20% bad observations.

The receiver operating characteristic curve (ROC) is a two-dimensional graphical illustration of the trade-off between the true positive rate (sensitivity) and false positive rate (1-specificity). The ROC curve illustrates the behaviour of a classifier without having to take into consideration the class distribution or misclassification cost. In order to compare the ROC curves of different classifiers, the area under the receiver operating characteristic curve (AUC) must be computed. The AUC statistic is similar to the Gini coefficient which is equal to $2 \times (AUC - 0.5)$. An example of an ROC curve is depicted in Fig. 1:

The diagonal line represents the trade-off between the sensitivity and (1-specificity) for a random model, and has an AUC of 0.5. For a well performing classifier the ROC curve needs to be as far to the top left-hand corner as possible. In the example shown in Fig. 1, the classifier that performs the best is the ROC_1 curve.

4.3. Parameter tuning and input selection

The linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and logistic regression (LOG) classification techniques require no parameter tuning. The LOG model was built in SAS using proc logistic and using a stepwise variable selection method. Both the LDA and QDA techniques were run in SAS using proc discrim. Before all the techniques were run, dummy variables were created for the categorical variables. The AUC statistic was computed using the ROC macro by DeLong, DeLong, and Clarke-Pearson (1988), which is available from the SAS website (http://.support.sas.com/kb/25/017.html).

For the LS-SVM classifier, a linear kernel was chosen and a grid search mechanism was used to tune the hyper-parameters. For the LS-SVM, the LS-SVMlab Matlab toolbox developed by Suykens et al. (2002) was used.

The NN classifiers were trained after selecting the best performing number of hidden neurons based on a validation set. The neural networks were trained in SAS Enterprise Miner using a logistic hidden and target layer activation function.

The confidence level for the pruning strategy of C4.5 was varied from 0.01 to 0.5, and the most appropriate value was selected for each data set based on validation set performance. The tree was built using the Weka (Witten & Frank, 2005) package.

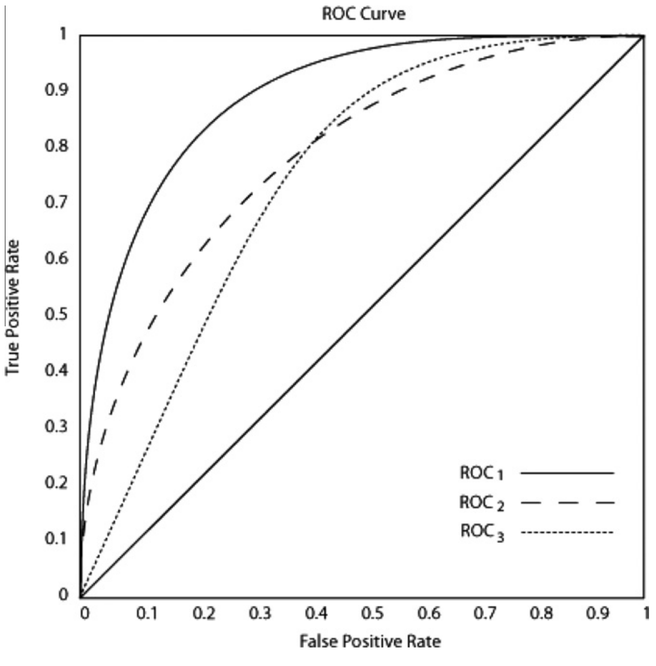

Fig. 1. Example ROC curve.

Two parameters have to be set for the Random Forests technique: these are the number of trees and the number of attributes used to grow each tree. A range of [10,50,100,250,500,1000] trees has been assessed, as well as three different settings for the number of randomly selected attributes per tree ($[0.5, 1, 2].\sqrt{M}$), whereby M denotes the number of attributes within the respective data set (Breiman, 2001). As with the C4.5 algorithm, Random Forests were also trained in Weka (Witten & Frank, 2005), using 10-fold cross-validation for tuning the parameters.

The k-Nearest Neighbours technique was applied for both $k = 10$ and $k = 100$, using the Weka (Witten & Frank, 2005) IBk classifier. For the gradient boosting classifier a partitioning algorithm was used as proposed by Friedman (2001). The number of iterations was varied in the range [10,50,100,250,500,1000], with a maximum branch size of two selected for the splitting rule (Friedman, 2001). The gradient boosting node in SAS Enterprise Miner was used to run this technique.

4.4. Statistical comparison of classifiers

We used Friedman's test (Friedman, 1940) to compare the AUCs of the different classifiers. The Friedman test statistic is based on the average ranked (AR) performances of the classification techniques on each data set, and is calculated as follows:

$$\chi_F^2 = \frac{12D}{K(K+1)}\left[\sum_{j=1}^{K} AR_j^2 - \frac{K(K+1)^2}{4}\right], \quad \text{where} \quad AR_j = \frac{1}{D}\sum_{i=1}^{D} r_i^j. \qquad (11)$$

In (13), D denotes the number of data sets used in the study, K is the total number of classifiers and r_i^j is the rank of classifier j on data set i. χ_F^2 is distributed according to the Chi-square distribution with $K-1$ degrees of freedom. If the value of χ_F^2 is large enough, then the null hypothesis that there is no difference between the techniques can be rejected. The Friedman statistic is well suited for this type of data analysis as it is less susceptible to outliers (Friedman, 1940).

The post hoc Nemenyi test (Nemenyi, 1963) is applied to report any significant differences between individual classifiers. The Nemenyi post hoc test states that the performances of two or more classifiers are significantly different if their average ranks differ by at least the critical difference (CD), given by

$$CD = q_{\alpha,\infty,K}\sqrt{\frac{K(K+1)}{12D}}. \qquad (12)$$

In this formula, the value $q_{\alpha,\infty,K}$ is based on the studentised range statistic (Nemenyi, 1963). Finally, the results from Friedman's statistic and the Nemenyi post hoc tests are displayed using a modified version of Demšar (2006) significance diagrams (Lessmann, Baesens, Mues, & Pietsch, 2008). These diagrams display the ranked performances of the classification techniques along with the critical difference to clearly show any techniques which are significantly different to the best performing classifiers.

5. Results and discussion

The table on the following page (Table 3) reports the AUCs of all ten classifiers on the five credit scoring data sets at varying degrees of class imbalance. For each level of imbalance, the Friedman test statistic and corresponding p-value is shown. As these were all significant ($p < 0.005$) a post hoc Nemenyi test was then applied to each class distribution. The technique achieving the highest AUC on each data set is underlined as well as the overall highest ranked technique. Table 3 shows that the gradient boosting algorithm has the highest Friedman score (average rank (AR)) on two of the five different percentage class splits. However at the extreme class split (99% good, 1% bad) Random Forests provides the best average

Table 3
Area under the receiver operating characteristic curve (AUC) results on test set data sets.

	30% Bad Friedman test statistic = 31.86 ($p < 0.005$)						15% Bad Friedman test statistic = 29.23 ($p < 0.005$)						10% Bad Friedman test statistic = 26.37 ($p < 0.005$)					
	Bene1	Bene2	Germ	Aus	Behav	AR	Bene1	Bene2	Germ	Aus	Behav	AR	Bene1	Bene2	Germ	Aus	Behav	AR
LOG	79.6	78.7	76.7	90.6	63.4	5.4	79.4	78.0	74.0	91.8	67.8	4.5	78.1	78.8	76.6	50.0	65.4	4.4
C4.5	71.4	71.0	71.2	91.8	61.9	9.1	69.7	60.9	65.2	91.6	61.6	8.2	64.7	64.0	64.1	91.9	50.3	8.4
NN	78.6	78.1	72.7	92.1	72.1	5.7	75.5	77.6	70.1	92.1	70.0	5.7	75.1	76.4	72.4	89.7	68.8	5.8
Gradient boosting	78.2	81.2	77.2	94.9	72.1	3.7	79.8	80.3	75.0	94.8	70.7	2.3	78.0	80.2	75.3	93.8	63.3	3.2
LDA	79.2	78.0	79.1	94.4	75.6	3.8	78.6	77.4	76.0	93.8	76.6	3.2	77.9	77.3	74.2	94.5	70.1	3.2
QDA	75.4	73.7	71.8	85.5	63.0	8.5	68.4	72.5	59.7	65.4	51.4	9.2	67.2	70.8	52.8	84.9	50.7	8.4
Random forests	78.5	79.0	80.0	93.7	76.2	3.2	77.9	78.0	76.9	94.1	76.5	2.7	78.6	76.9	77.2	93.2	74.7	2.2
k-NN10	76.2	71.0	75.0	92.8	61.8	7.7	75.5	68.1	71.7	90.3	58.7	7.9	70.4	64.4	68.8	92.5	56.3	6.8
k-NN100	75.4	73.9	79.3	93.0	56.0	6.7	75.8	73.6	78.1	92.6	62.9	4.6	75.3	72.9	78.5	92.3	61.7	4.6
Lin LS-SVM	80.3	80.6	81.9	95.1	82.9	1.2	50.0	54.4	75.0	91.0	90.0	6.7	50.0	50.0	76.8	90.6	50.0	8.0
	5% Bad Friedman test statistic = 26.29 ($p < 0.005$)						2.5% Bad Friedman test statistic = 27.43 ($p < 0.005$)						1% Bad Friedman test statistic = 30.86 ($p < 0.005$)					
	Bene1	Bene2	Germ	Aus	Behav	AR	Bene1	Bene2	Germ	Aus	Behav	AR	Bene1	Bene2	Germ	Aus	Behav	AR
LOG	75.0	75.4	75.7	50.0	50.0	5.5	72.7	73.9	55.1	50.0	50.0	6.3	50.0	64.7	50.0	50.0	50.0	7.7
C4.5	58.6	64.9	56.5	75.4	55.0	7.6	65.8	67.9	61.4	58.7	53.9	7.0	50.0	55.5	64.2	50.0	50.0	6.9
NN	68.4	70.7	68.3	89.4	64.4	5.0	71.2	70.2	59.2	70.0	62.3	4.6	50.0	62.5	54.2	86.7	54.0	5.6
Gradient boosting	70.8	78.0	76.6	93.1	52.7	3.4	68.1	74.7	71.4	88.3	55.6	2.8	58.1	69.1	59.4	74.5	51.0	3.5
LDA	74.1	76.1	73.8	93.5	63.5	2.6	75.7	72.2	62.6	81.8	60.6	3.0	50.2	69.0	58.3	86.8	54.6	3.4
QDA	63.8	72.0	50.0	59.7	50.5	7.9	66.5	65.3	50.0	51.6	50.5	8.3	50.0	50.0	50.0	52.0	50.7	7.9
Random forests	73.2	75.8	75.2	93.2	63.1	3.2	69.2	71.3	69.1	87.9	68.7	3.0	61.9	67.4	67.1	90.1	60.0	1.6
k-NN10	65.2	62.0	67.1	88.6	53.5	7.0	59.0	56.3	59.3	72.8	54.7	7.0	52.5	52.3	54.8	67.2	50.0	6.5
k-NN100	74.7	71.3	75.8	92.3	59.8	3.6	70.6	68.8	69.3	87.8	58.3	3.8	67.2	63.2	63.6	90.0	51.0	3.1
Lin LS-SVM	50.0	50.0	50.0	87.8	50.0	9.2	50.0	50.0	50.0	65.2	50.0	9.2	50.0	50.0	50.0	50.0	50.0	8.8

ranking across the five data sets (Random Forests also ranks first on the 10% data set).

In the majority of the class splits, the AR of the QDA and Lin LS-SVM classifiers are statistically worse than the AR of the Random Forests classifier at the 5% critical difference level ($\alpha = 0.05$), as shown in the significance diagrams included next. Note that, even though the differences between the classifiers are small, it is important to note that in a credit scoring context, an increase in the discrimination ability of even a fraction of a percent may translate into significant future savings (Henley & Hand, 1997).

The following significance diagrams display the AUC performance ranks of the classifiers, along with Nemenyi's critical difference (CD) tail. The CD value for all the following diagrams is equal to 6.06. Each diagram shows the classification techniques listed in ascending order of ranked performance on the *y*-axis, and the classifier's mean rank across all five data sets displayed on the *x*-axis. Two vertical dashed lines have been inserted to clearly identify the end of the best performing classifier's tail and the start of the next significantly different classifier.

The first significance diagram (see Fig. 2) displays the average rank of the classifiers at the original class distribution of a 70% good, 30% bad split:

At this original 70/30% split, the linear LS-SVM is the best performing classification technique with an AR value of 1.2. This diagram clearly shows that the *k*-NN10, QDA and C4.5 techniques perform significantly worse than the best performing classifier with values of 7.7, 8.5 and 9.1 respectively.

The following significance diagram displays the average rank of the classifiers at an 85% good, 15% bad class split:

At the level where only 15% of the data sets are bad observations, it is shown in the significance diagram that gradient boosting becomes the best performing classifier (see Fig. 3). The gradient boosting classifier performs significantly better than the quadratic discriminant analysis (QDA) classifier. From these findings we can make a preliminary assumption that when a larger class imbalance is present, the QDA classifier remains significantly different to the gradient boosting classifier. All the other techniques used are not significantly different.

At a 90% good, 10% bad class split the significance diagram shown in Fig. 4 indicates that the C4.5 and QDA algorithms are significantly worse than the random forests classifier. It can be noted that the Linear LS-SVM classifier however is progressively becoming less powerful as a large class imbalance is present (see Fig. 5).

The final split, displaying a 99% good, 1% bad class split, indicates that, at the most extreme class distribution analysed, two classification techniques are significantly worse (Lin LS-SVM and QDA). This displays an interesting finding that at the extreme split, LOG is now close to being significantly worse than the Random Forests algorithm. The logistic regression technique therefore shows limited power in correctly classifying observations where only a small number of bad observations exist. It can also be concluded that the random forests classifier performs surprisingly well given a large class imbalance.

In summary, when considering the AUC performance measures, it can be concluded that the gradient boosting and random forest classifiers yield a very good performance at extreme levels of class imbalance, whereas the Lin LS-SVM sees a reduction in performance as a larger class imbalance is introduced. However, the simpler, linear classification techniques such as LDA and LOG also give a relatively good performance, which is not significantly different from that of the gradient boosting and random forest classifiers. This finding seems to confirm the suggestion made in Baesens et al. (2003) that most credit scoring data sets are only weakly non-linear. However, techniques such as QDA, C4.5 and *k*-NN10 perform significantly worse than the best performing classifiers at each percentage reduction. The majority of classification tech-

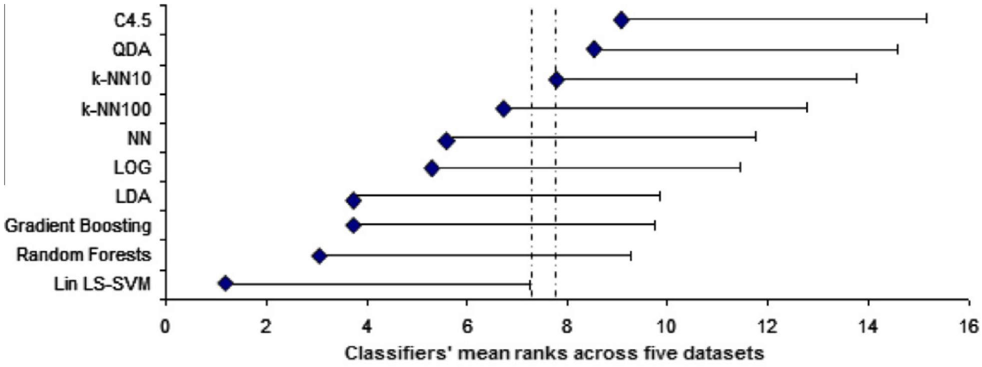

Fig. 2. AR comparison at a 70/30% split of good/bad observations.

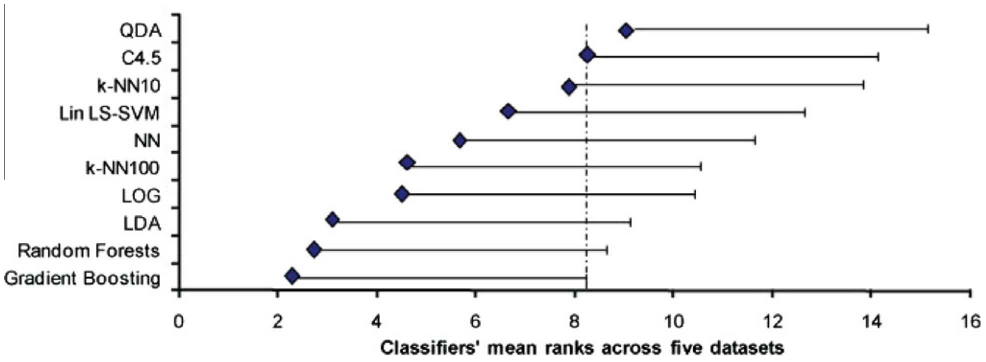

Fig. 3. AR comparison at an 85/15% split of good/bad observations.

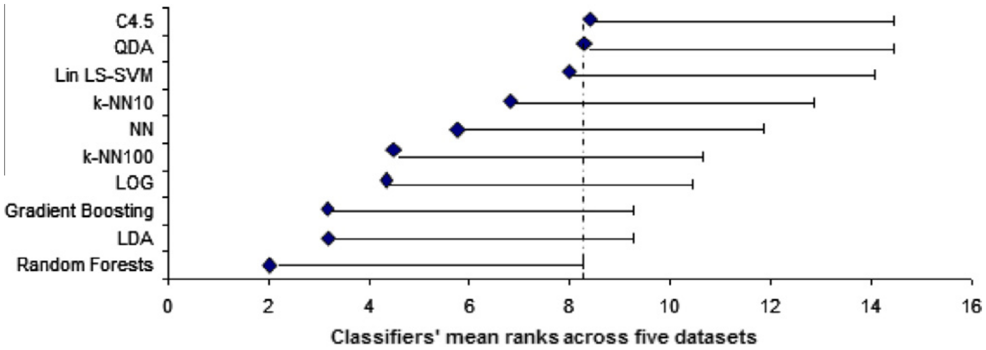

Fig. 4. AR comparison at a 90/10% split of good/bad observations.

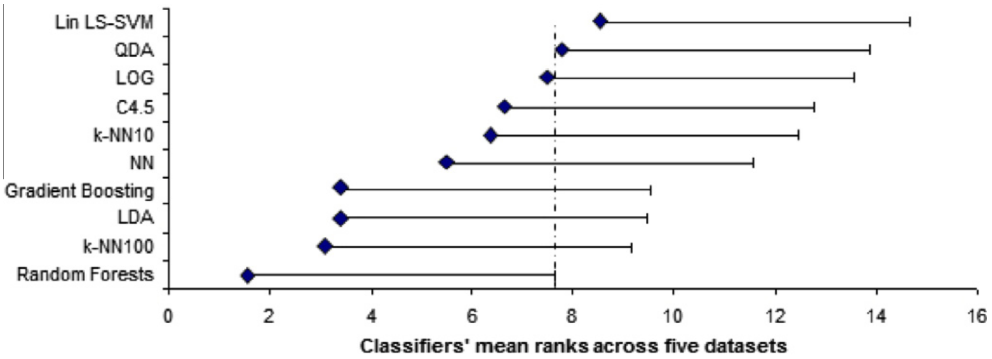

Fig. 5. AR comparison at a 99/1% split of good/bad observations.

niques yielded classification performances that are quite competitive with each other.

6. Conclusions and recommendations for further work

In this comparative study we have looked at a number of credit scoring techniques, and studied their performance over various class distributions in five real-life credit data sets. Two techniques that have yet to be fully researched in the context of credit scoring, i.e., gradient boosting and random forests, were also chosen to give a broader review of the techniques available. The classification power of these techniques was assessed based on the area under the receiver operating characteristic curve (AUC). Friedman's test and Nemenyi's post hoc tests were then applied to determine whether the differences between the average ranked performances of the AUCs were statistically significant. Finally, these significance results were visualised using significance diagrams for each of the various class distributions analysed.

The results of these experiments show that the gradient boosting and random forest classifiers performed well in dealing with samples where a large class imbalance was present. It does appear that in extreme cases the ability of random forests and gradient boosting to concentrate on 'local' features in the imbalanced data is useful. The most commonly used credit scoring techniques, linear discriminant analysis (LDA) and logistic regression (LOG), gave results that were reasonably competitive with the more complex techniques and this competitive performance continued even when the samples became much more imbalanced. This would suggest that the currently most popular approaches are fairly robust to imbalanced class sizes. On the other hand, techniques such as QDA and C4.5 were significantly worse than the best performing classifiers. It can also be concluded that the use of a linear kernel LS-SVM would not be beneficial in the scoring of data sets where a very large class imbalance exists.

Further work that could be conducted, as a result of these findings, would be to firstly consider a stacking approach to classification through the combination of multiple techniques. Such an approach would allow a meta-learner to pick the best model to classify an observation. Secondly, another interesting extension to the research would be to apply these techniques on much larger data sets which display a wider variety of class distributions. It would also be of interest to look into the effect of not only the percentage class distribution but also the effect of the actual number of observations in a data set.

Finally, as stated in the literature review section of this paper, there have been several approaches already researched in the area of over-sampling techniques to deal with large class imbalances. Further research into this and their effect on credit scoring model performance would be beneficial.

Acknowledgements

The authors of this paper would like to thank the EPSRC and SAS UK for their financial support to Iain Brown.

References

Altman, E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance, 23*(4), 589–609.

Altman, E. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian Experience). *Journal of Banking & Finance, 18*(3), 505–529.

Arminger, G., Enache, D., & Bonne, T. (1997). Analyzing credit risk data: A comparison of logistic discrimination, classification tree analysis, and feed forward networks. *Computational Statistics, 12*, 293–310.

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.

Batista, G. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter, 6*(1), 20–29.

Benjamin, N., Cathcart, A., & Ryan, K. (2006). *Low default portfolios: A proposal for conservative estimation of default probabilities*. Discussion Paper. Financial Services Authority.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Chatterjee, S., & Barcun, S. (1970). A nonparametric approach to credit screening. *Journal of the American Statistical Association, 65*(329), 50–154.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research, 16*, 321–357.

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics, 44*(3), 837–845.

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

Desai, V. S., Crook, J. N., & Overstreet, G. A. Jr., (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics, 11*(1), 86–92.

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232.

Friedman, J. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367–378.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning, data mining, inference, and prediction*. New York: Springer.

Henley, W. E., & Hand, D. J. (1997). Construction of a *k*-nearest neighbour credit-scoring system. *IMA Journal of Management Mathematics, 8*(4), 305–321.

Hosmer, D. W., & Stanley, L. (2000). *Applied logistic regression* (2nd ed.). Chichester, New York: Wiley.

Japkowicz, N. (2000). Learning from imbalanced data sets: A comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets* (Vol. 6, pp. 10–15).

Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking classification models for software defect prediction: A proposed framework and novel findings. *IEEE Transactions on Software Engineering, 34*(4), 485–496.

Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Ph.D. Thesis. Princeton University.

Provost, F., Jensen, D., & Oates, T. (1999). Efficient progressive sampling. In *Proceedings of the fifth international conference on knowledge discovery and data mining*. ACM Press.

Quinlan, J. R. (1993). *C4.5 programs for machine learning*. San Mateo, CA: Morgan Kaufmann.

Steenackers, A., & Goovaerts, M. J. (1989). A credit scoring model for personal loans. *Insurance: Mathematics and Economics, 8*(1), 31–34.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. Singapore: World Scientific.

Van Der Burgt, M. (2007). *Calibrating low-default portfolios, using the cumulative accuracy profile*. ABN AMRO.

Weiss, G. M., & Provost, F. J. (2003). Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research, 19*, 315–354.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research, 27*(11–12), 1131–1152.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behavior. *Journal of Financial and Quantitative Analysis, 15*, 757–770.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Yang, Y. (2007). Adaptive credit scoring with kernel learning methods. *European Journal of Operational Research, 183*(3), 1521–1536.

Yobas, M. B., Crook, J. N., & Ross, P. (2000). Credit scoring using neural and evolutionary techniques. *IMA Journal of Management Mathematics, 11*(2), 111–125.