Appl Intell (2007) 27:29–37
DOI 10.1007/s10489-006-0028-9

A boosting approach for corporate failure prediction

Esteban Alfaro Cortés · Matías Gámez Martínez · Noelia García Rubio

Received: 6 July 2006 / Accepted: 25 October 2006 / Published online: 8 December 2006

Abstract Predicting corporate failure is an important management science problem. This is a typical classification question where the objective is to determine which indicators are involved in the failure/success of a corporation. Despite the importance of this problem, until now only classical machine learning tools have been considered to tackle this classification task. The objective of this paper is twofold. On the one hand, we introduce novel discerning measures to rank independent variables in a generic classification task. On the other hand, we apply boosting techniques to improve the accuracy of a classification tree. We apply this methodology to a set of European firms, considering the usual predicting variables such as financial ratios, as well as including novel variables rarely used before in corporate failure prediction, such as firm size, activity and legal structure. We show that our approach decreases the generalization error about thirty percent with respect to the error produced with a classification tree. In addition, the most important ratios deal with profitability and indebtedness, as is usual in failure prediction studies.

Keywords Ensemble classifiers · Boosting · Corporate failure prediction

E. A. Cortés (✉) · M. G. Martínez · N. G. Rubio
Economic and Business Sciences Faculty of Albacete, Castilla-La Mancha University, Plaza de la Universidad, 1. 02071 Albacete, Spain
e-mail: Esteban.Alfaro@uclm.es

M. G. Martínez
e-mail: Matias.Gamez@uclm.es

N. G. Rubio
e-mail: Noelia.Garcia@uclm.es

1 Introduction

As in any classification task, a set of n observations is initially given and noted as $T_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ where each X_i is a p-dimensional vector whose components are the values of the i-th observation in each of the p features, i.e. $X_i = \{X_{i1}, X_{i2}, \ldots, X_{ip}\}$ and Y_j, for $j = 1, 2, \ldots, k$, is the observation class label. On the basis of the training set, a classifier is constructed, generally as a function of the p features, $C(X_i) = f(X_i)$. This function is used to predict the Y class while minimizing the prediction error.

In classification problems, an ensemble of classifiers can be used to increase prediction accuracy by aggregating the predictions of several classifiers, and the words aggregation, combination and ensemble are synonymous in this field of research. The classifier which is built by combining various classifiers is called an ensemble of classifiers. There are several alternatives and the first is to build different classifiers from the data set and then combine them by simple vote or linear functions. Another perhaps more sophisticated possibility consists in applying the same classification method on modified versions of the learning set. Some of these techniques are relatively new and have been studied closely in recent years and of these bagging and boosting methods deserve special mention [1–3].

Predicting corporate failure is an important management science problem. The lack of a unified theory on corporate failure has meant that most studies dealing with distress prediction have focused on increasing the accuracy of the model and have not always paid enough attention to model interpretation. This is clearly of crucial importance in failure prediction as the firm must make appropriate decisions. In this research, a new method is proposed for predicting corporate failure and as far as we are aware, this is the first study to apply boosting techniques to corporate failure prediction. To

illustrate its usefulness, we will apply this method on a selection of Spanish companies, and in order to ensure that these results are general and can be projected to other European countries and the United States, we will use financial ratios that have proved significant for predicting business failure in previous studies (e.g. Frydman [4]). We will also propose a new measure for the importance of variables to facilitate model interpretation. This measure takes into account how often variables are actually used in the individual trees and on the basis of this measure, the variables can be ranked in terms of importance.

The following factors should be taken into account within the empirical application:

- Acquired and dissolved firms are also considered rather than only bankrupt and temporary receivership firms (which is usual in corporate failure prediction literature).
- Qualitative variables such as the firm size, activity, and legal structure are included as predictors in addition to the usual financial ratios.
- The boosting method is applied to the corporate prediction, analyzing the extent to which this methodology is suitable for the subject.

In Section 2 of this paper, we present the boosting method included in the study with a discussion of how it works in practice and we describe the main algorithm used. The following section introduces the failure prediction problem and the data used in the analysis. The classification results are then presented and the well-known classification tree model is compared with the new boosting classifier. Finally, following on from the empirical analysis, we present our conclusions.

2 Boosting

As mentioned in the Introduction, a classifier system builds a model which is able to predict the class of a new observation given a data set. The accuracy of the classifier will depend on the quality of the method used and the difficulty of the specific application. If the obtained classifier achieves a better accuracy than the default rule, then the classification method has found some structure in the data enabling it to do so. Boosting [1] is a method that makes maximum use of a classifier by improving its accuracy. The classifier method is therefore used as a subroutine to build an extremely accurate classifier in the training set.

Boosting applies the classification system repeatedly on the training data, but with each application, the learning attention is focused on different examples of this set. Once the process has finished, the single classifiers obtained are combined into a final, highly accurate classifier in the training set. The final classifier therefore usually achieves a high degree of accuracy in the test set as various authors have shown both theoretically and empirically [5–15].

Even though there are several versions of the boosting algorithm [5], the most widely used is the one by Freund and Schapire [1] which is known as Adaboost. For simplification purposes, it can be assumed that there are only two classes without loss of generality. A training set is given $T_n = \{(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)\}$ where Y takes values $\{-1, 1\}$. The weight $w_b(i)$ is assigned to each observation X_i and is initially set to $1/n$. This value will be updated after each step. A basic classifier denoted $C_b(X_i)$ is built on this new training set and is applied to each training example. The error of this classifier is represented by ε_b and is calculated as

$$\varepsilon_b = \sum_{i=1}^{n} w_b(i)\xi_b(i) \quad \text{where} \quad \xi_b(i) = \begin{cases} 0 & C_b(x_i) = y_i \\ 1 & C_b(x_i) \neq y_i \end{cases} \tag{1}$$

From the error of the classifier in the b-th iteration the constant α_b is calculated and this value is used to update the weights. More specifically, according to the authors mentioned above $\alpha_b = \ln(1 - \varepsilon_b/\varepsilon_b)$ and the new weight for the $b + 1$-th iteration will be

$$w_{b+1}(i) = w_b(i)\, \exp(\alpha_b \xi_b(t)) \tag{2}$$

The calculated weights are then normalized so that they add up to one. Accordingly, $\varepsilon_b = 0.5 - \gamma_b$, where γ_b shows the advantage of the basic classifier of the b-th step over the default rule in the worst case, where both classes have the same a priori probability of 0.5.

The following example shows how weights are updated depending on the error level in the second step, $b = 2$. For this, two values are selected near the limits of the ε_b domain (0 and 0.5). If $\varepsilon_b = 0.499$, then $\alpha_b = 0.004$ and the new weight for the second step is $w_2(i) = 1/n\ \exp(-0.004\xi_b(i))$. Therefore, if the i-th observation is wrongly classified, its weight $w_2(i) = 1/n \cdot 1.004$, whereas if it is correctly classified its weight will be reduced during normalization. Assuming that $\varepsilon_b = 0.001$, then $\alpha_b = 6.907$ and the weight of a wrongly classified example will be $w_2(i) = 1/n \cdot 999$, whereas the weight of the rightly classified examples will be reduced in the normalization.

Table 1 displays the weights for the second iteration after the normalization process for a set of 1000 observations and shows how the weight of the wrongly classified observations increases and the weight of the correctly classified ones decreases, forcing the single classifier built in the following iteration to focus on the hardest examples. In addition, the differences when the weights are updated are greater when the error of the single classifier is small since more importance

Table 1 Example of the weight updating process in Adaboost

n	Initial weight	Error	Alpha	Wrongly classified	Weight (2)	Weight (2) normalized
1000	0.001	0.499	0.004	1	0.001004	0.001002
1000	0.001	0.499	0.004	0	0.001000	0.000998
1000	0.001	0.001	6.907	1	0.999000	0.500000
1000	0.001	0.001	6.907	0	0.001000	0.000501

Table 2 Adaboost algorithm (Freund and Schapire [1])

1. Start with $w_i^1 = 1/n, i = 1, 2, \ldots, n$.
2. Repeat for $b = 1, 2, \ldots, B$
   (a) Fit the classifier $C_b(x) \in \{-1, 1\}$ using weights w_i^b on T^b.
   (b) Compute: $\varepsilon_b = \sum_{i=1}^{n} w_i^b \xi_b(i)$ and $\alpha_b = \ln(1 - \varepsilon_b/\varepsilon_b)$
   (c) Update the weights $w_i^{b+1} = w_i^b \exp(\alpha_b \xi_b(i))$ and normalize them.
3. Output the final classifier

$$C(x) = \text{sign}\left(\sum_{b=1}^{B} \alpha_b C_b(x)\right)$$

is given to the few mistakes mentioned when the classifier achieves a high level of accuracy. The alpha constant can therefore be interpreted as a *learning rate* which is calculated as a function of the error made on each iteration. This constant is also used in the final decision rule giving more importance to the individual classifiers that made a smaller error.

This process is repeated in every step for $b = 1, 2, 3, \ldots, B$. Finally, the ensemble classifier is built as a linear combination of the single classifiers weighted by the corresponding constant α_b.

$$C(x) = \text{sign}\left(\sum_{b=1}^{B} \alpha_b C_b(x)\right) \tag{3}$$

Table 2 summarises the Adaboost algorithm.

Freund and Schapire [2] show that when the number B of iterations is increased, the training error level of the Adaboost classifier exponentially tends to zero. They also demonstrate that the generalization or true error (ε_R) of the final classifier $C_F(x)$ has an upper limit which depends on the training or apparent error (ε_A), the size of the training set (n), Vapnik–Chervonenkis's dimensionality coefficient of the parametric space of basic classifiers (d) and the number of iterations B in boosting (number of combined single classifiers)

$$\hat{\varepsilon}_R = \hat{\varepsilon}_A + \hat{\theta}\sqrt{\frac{Bd}{n}} \tag{4}$$

Although the generalization error of the final classifier may be reduced by increasing the size of the training set, the generalization error will increase when the number of single classifiers included increases. This is due to overfitting the classifier (a classifier is said to be overfitted when it is too closely adjusted to the training set, thereby losing its generalization capacity on the total population, and will therefore be inaccurate when classifying new examples).

3 Problem description

Predicting corporate failure is an important management science problem and the main goal of corporate failure prediction is to differentiate between those firms with a high probability of distress in the future from healthy firms. In other words, a model is built to forecast the moment of distress so that the firm's economic agents may make suitable decisions. In order to be able to predict failure, it is essential to have access to information about the company's situation. This information is basically given by financial ratios but additional information (e.g. activity, company size, age, etc.) should also be taken into account.

Predicting corporate failure is not a new research field and many studies have dealt with this problem since 1966. It is therefore interesting to study the state of the art of corporate failure prediction. There is no doubt that the pioneering failure prediction studies were provided by Beaver [16] at a univariate level and Altman [17] with the application of multivariate analysis. Subsequently, and basically to overcome the restrictive statistical requirements of normality for the predictor variables and equality for the variance-covariance group matrices, logit and probit models were also applied (Ohlson [18] and Zmijewski [19]). Classification trees or recursive partitioning proved useful in studies by Frydman et al. [4]. More recently, artificial neural networks have been introduced as a powerful approach to this task (Wilson and Sharda [20]). Care should be taken when comparing the prediction ability of alternative techniques in accounting because of the different starting conditions in alternative studies.

Laitinen [21] presents an interesting collection of the main failure prediction studies which are grouped according to the classification method used for prediction. Appendix A lists some of the pioneering studies so as to enable comparison with the results of this paper. However, owing to differences in the initial conditions, the precautions mentioned above should be taken into account.

Although there is a general consensus on the importance of failure prediction, there is not the same degree of agreement on the definition of corporate failure (i.e. when a company is considered to have failed). From a global perspective, a firm will have failed if it does not achieve its goals, especially those relating to profitability, solvency, and survival. In line with this definition, a wider concept of corporate failure is used in this research. Although in Spanish failure prediction studies, only bankruptcy and temporary receivership companies have traditionally been considered due to corresponding legislation, in this study, acquired and dissolved firms are also included as failed firms. Acquired firms are considered to have failed because the company loses its identity, sometimes as a result of poor management, although this question should be studied thoroughly in future research.

4 Data description

The companies in the sample were selected from the SABI database of Bureau Van Dijk (BVD), one of Europe's leading publishers of electronic business information databases and provider of the Wharton Research Data Services. SABI-covers all the companies whose accounts are placed on the Spanish Mercantile Registry. In the case of failed firms, firms which had failed during the period 2000–2003 were selected, but with the additional requirement that full information be provided about all the variables at the moment of failure and the previous year. It is usual in failure prediction studies to select failed firms from various years in order to collect a higher sample size. There were therefore firms that had failed in years 2000, 2001, 2002 and 2003 so the information on variables should be understood in relative terms with respect to the moment of failure (t) with the previous year being $t-1$.

Healthy firms, on the other hand, were selected from active companies at the end of 2003 with full data for 2003 and 2002. In this case, a second requirement was added: any firm with constantly negative profits during the last three years would be rejected, the reason being that even though they were still active in December 2003, they would soon enter a state of failure if they kept making a loss.

Within these requirements, 1365 firms were randomly selected for each group (failed/healthy), obtaining 2730 observations for the total set. Instead of pairing the failed/healthy firms by sector or size, the following variables were used as predictors in the selection process: the sector as a qualitative variable with ten categories using the National Classification of Economic Activities (NACE-93 digit-1 level), and the size using the natural logarithm of Total Assets as a proxy variable. The legal structure was also used as a categorical predictor with three options: public corporations, limited corporations, and other corporations.

In addition, fourteen accounting-based ratios were included in the initial data set. In failure prediction studies, financial ratios are usually selected on the basis of three criteria: they should be commonly used in failure prediction literature, the information needed to calculate these ratios should be available, and finally, the researchers' own decisions based on their experience in previous studies or on the basis of the preliminary trials. The same criteria were followed in this study. Eighteen predictor variables were therefore used for each company with information from the year prior to the moment of failure and these variables are listed in Appendix B.

5 Experimental results

In order to make the techniques used in this paper available to the statistical community who are usually involved in this particular decision problem, we have implemented them in the R statistical program. This consists of a series of packages for data manipulation, calculus, and graphics [22]. Among other characteristics, it has a well developed and effective programming language (R language). The R program has much in common with the well-known *S*-Plus program, but with one difference being that R is a freely available software program which can be downloaded from http://cran.r-project.org/.

Classification trees can be applied with R using the *rpart* library. The R program has a base environment, with a few statistical, mathematical, and graphical utilities. More sophisticated techniques can be added using packages which are available on the CRAN website mentioned above, e.g. the rpart library which allows classification trees to be applied (as shown below).

5.1 Corporate failure prediction through classification trees

In this paper, the same failure prediction problem is solved using two different classification methods in order to compare their classification accuracy in this task. To estimate the real accuracy, the total initial sample of 2730 Spanish companies is divided into two sets: ninety percent are used as a training set to build the classifier, and the rest are hidden from the classification method and are presented as new data to check the prediction ability. The training set therefore comprises 1228 healthy firms and a further 1228 failed firms

(2456 firms represent 90% of the total set). The test set consists of 274 firms, with an equal number of healthy and failed firms (10% of the total). As mentioned before, 18 predictor variables are used for each company with information for the year prior to the moment of failure.

Classification trees are a non-linear and non-parametric classification method [23], and they have the same structure as a real tree with nodes, branches and leaves. The initial set is called the root node and it is recursively split into mutually exclusive subsets or nodes. A test is used on each node to split it and to increase the homogeneity of the different subsets. In binary trees, only two branches can come up from each node. On each split, the variable that achieves a higher separation between classes is selected. When a stop criterion is reached, the majority class is assigned to the example in this node and this is called a terminal node or leaf. Since the nature of the information is not relevant, classification trees are an excellent method for qualitative variables.

A pruned tree is trained using the *1-SE rule*. This rule selects the smallest tree with a cross validation error equal to or less than the minimum error plus a standard deviation, and the results are displayed exactly as the R program presents them.

```
> prune(sabi.rpart, cp = 0.01) ->sabi.prune
> printcp(sabi.prune)

Classification tree:
rpart(formula = ESTADO ~ ., data = sabi [ind, ],
method = "class", cp = 0, minsplit = 1,
maxdepth = 30)

Variables actually used in tree construction:
[1] CF.PT1 PE.PT1

Root node error: 1228/2456 = 0.5

n = 2456

        CP nsplit rel error  xerror    xstd
1 0.767101      0   1.00000 1.04560 0.020157
2 0.012215      1   0.23290 0.23453 0.012984
3 0.010000      2   0.22068 0.23534 0.013004
```

The tree obtains an error of 11.03% in the training set and its confusion matrix is shown immediately after.

```
> sabi.predrpart < − predict(sabi. prune,
newdata = sabi[ind,], type = "class")
>  1-sum(sabi.predrpart == sabi[ind, "ESTADO"])
/length(sabi[ind, "ESTADO"]) [1] 0.1103420
> table(sabi.predrpart, sabi$ESTADO [ind],
dnn = c("Predicted class", "Observed class"))
                    Observed class
  Predicted class   Failed   Healthy
           Failed     1225       268
          Healthy        3       960
```

The error test is 9.124%. The two failed firms which are classified as healthy are shown in the confusion matrix for this set.

```
> sabi.predrpart < − predict(sabi.prune,
newdata = sabi[-ind,], type = "class")

> 1-sum(sabi.predrpart == sabi[-ind, "ESTADO"])
/length(sabi[-ind,"ESTADO"]) [1] 0.09124088

> table(sabi.predrpart, sabi$ESTADO [-ind],
dnn = c("Predicted class", "Observed class"))
                Observed class
Predicted class   Failed   Healthy
     Failed         135        23
    Healthy           2       114
```

Figure 1 shows the tree structure and can be seen below in greater detail.

```
> sabi.prune
n = 2456

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 2456 1228 Failed  (0.5000000 0.5000000)
2) CF.PT1 < 0.4915537 1514 286 Failed
   (0.8110964 0.1889036)
     4) PE.PT1 < 2.536775 1493 268 Failed
        (0.8204956 0.1795044) *
    5) PE.PT1 >= 2.536775 21 3 Healthy
       (0.1428571 0.8571429) *
3) CF.PT1 >= 0.4915537 942 0 Healthy
(0.0000000 1.0000000) *
```

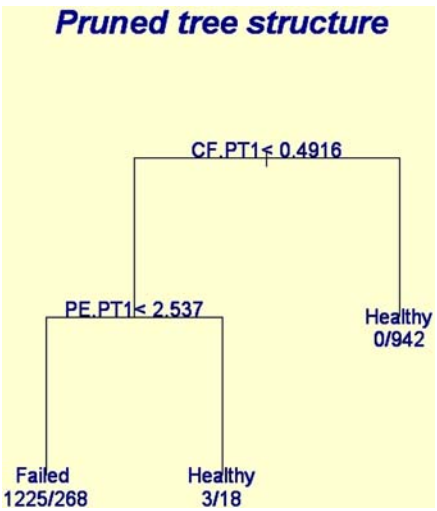

Fig. 1 Pruned tree structure

In this case, the pruned tree only uses two ratios (CF.PT1 and PE.PT1) to discriminate between healthy and failed firms, and both ratios deal with the indebtedness of the firm. A test is set at each node and the examples which satisfy the test are assigned to the left-hand branch while the rest are assigned to the right-hand one. A label (i.e. healthy or failed) is assigned to each leaf and the number of failed and healthy firms present on the leaf is shown. For instance, on the left-hand leaf, the class is failed and there are 1225 failed firms and 268 healthy firms.

5.2 Corporate failure prediction through the boosting method

Although boosting methods can use any sort of classification system as the individual classifier, decision trees are used in this application for the task. There are basically three reasons for this: decision trees are used in most boosting applications, they achieve good results, and classification trees easily handle qualitative variables.

Three functions were developed in R language: the first trains the Adaboost classifier and assigns a class to the examples of the training set, the second uses a previously trained Adaboost classifier to predict the classes of the cases in the new data set, and the third enables cross validation to be applied in order to estimate the error of an Adaboost classifier. As in any R function, there are a few initial arguments to be set such as the name of the data frame where the data are stored or the name of the variable that contains the observation class and the predictor variables, the number of individual trees to be used, and the size of these trees.

Once the functions described above have been implemented, a boosting classifier is built with 100 trees that have been pruned using cp = 0.01 to limit the size of the individual tree in each boosting epoch. Since the test error is reduced to 6.569%, there is a reduction of 28% compared with the individual tree test error, which is 9.124%. In addition, if the confusion matrix is analyzed, it can be seen that most of these errors are made because a healthy firm is classified as failed. The following charts show the error in the training and test sets.

```
> sabi.boosting <- Adaboost(ESTADO ~ .,
data = sabi[ind,], mfinal = 100,
boos = T,cp = 0.01)

> table(sabi.boosting$clase, sabi$ESTADO[ind],
dnn = c("Predicted class", "Observed class"))

          Observed class
Predicted class   Failed   Healthy
Failed              1204       196
Healthy               24      1032

>1-sum(sabi.boosting$clase == sabi $ESTADO[ind])
/length(sabi$ESTADO[ind]) [1] 0.08957655
```

```
> predict.boosting(sabi.boosting,
newdata = sabi[-ind,]) -> sabi. predboost
> sabi.predboost[-1]
$"confussion"
                Observed class
Predicted class   Failed   Healthy
     Failed          135        16
    Healthy            2       121
$error
[1] 0.06569343
```

In order to ensure that comparison between individual trees and the boosting ensemble does not happen by chance, we use five repetitions of 10-fold cross-validation (Opitz and Maclin [24]). The entire set (2730 firms) is used for each 10-fold cross-validation experiment. The error rate is averaged and is shown in Table 3 together with the standard deviation of the error. Adaboost also reduces the error of the individual tree when a cross-validation analysis is developed, decreasing the average error by 5.24% compared with the individual tree. The average error of cross validation is undoubtedly useful for generalizing comparisons between two classification methods, but when the main goal is to find the best solution for a specific problem, the best classifier will be taken rather than the average one. In our case, since we are looking for the best way to predict failure, we select the Adaboost classifier which obtains a test error of 6.569%.

The Adaboost function allows the relative importance of the predictor variables to be quantified. This is a really important advantage because it is difficult to interpret the hundreds or thousands of trees used in the boosting ensemble. This measure takes into account how often each variable is selected to realize a split. It is logical to consider that the more important variables will be used in a greater number of splits than the less important ones. Table 4 shows all

Table 3 10-fold cross-validation

Error	Individual tree	Adaboost
Average	0.11436	0.10842
Standard deviation	0.00142	0.0015

Table 4 Relative importance of variables

Variable	Relative importance	Variable	Relative importance
CF.PT1	15.1	lnAC1	3.91
PE.PT1	12.5	T.PC1	3.65
BAI.FP1	11.2	IN.AT1	3.39
BAI.AT1	10.16	V.AC1	2.86
V.AT1	7.03	lnAT1	2.6
NACE1	5.73	FM.AT1	2.34
V.FP1	5.73	FM.V1	1.56
T.AT1	4.69	AC.PC1	1.56
AC.AT1	4.69	JURIDICA	1.3

variables arranged from the greatest to least relative importance. In this case, the most important ratios are CF.PT1, PE.PT1, BAI.FP1 and BAI.AT1 with values at this measure of 15.1, 12.5, 11.2, and 10.2%, respectively. Those variables which are different from financial ratios (NACE1, lnAC, lnAT and Juridica) have an interesting contribution of 13.5% in total.

The most important ratios in the analysis are the capacity for debt repayment, level of indebtedness, economic profitability, and financial profitability. The first of these ratios measures the capacity of the firm to repay a debt according to the generated resources, which are the sum of the earnings before interest and taxes plus the consumption of fixed assets and the expected depreciation of the fixed or current assets (provisions). The PE.PT1 ratio considers the weight of the indebtedness in the financial structure, showing that firms with lower levels of indebtedness are more able to apply for new external financing sources and therefore have greater possibilities of survival.

The other two variables involve the corporation's efficiency in the application of equity capital and assets, measuring them by means of the weight of generated earnings before interest and taxes on both accounting magnitudes. The most efficient firms in both aspects will undoubtedly have a greater likelihood of being classified as healthy.

In boosting literature, the concept of margin [25] is important. The margin for an object is intuitively related to the certainty of its classification and is calculated as the difference between the support of the correct class and the maximum support of an incorrect class. For q classes, the margin of an example x is calculated using the degree of support of the different classes $\mu_j(x)$, $j = 1, 2, \ldots, q$ as

$$m(x) = \mu_k(x) - \max_{j \neq k} \mu_j(x) \tag{5}$$

where k is the correct class of x and $\sum_{j=1}^{q} \mu_j(x) = 1$.

All the wrongly classified examples will therefore have negative margins and those correctly classified will have positive margins. Correctly classified observations with a high degree of confidence will have margins which are close to one whereas examples with an uncertain classification will have small margins and margins close to zero. Since a small margin is an instability symptom in the assigned class, the same example could be assigned to different classes by similar classifiers. For visualization purposes, Kuncheva [26] uses margin distribution graphs showing the cumulative distribution of the margins for a given data set. The x-axis is the margin (m) and the y-axis is the number of points where the margin is less than or equal to m. If all the training points have been classified correctly, there will only be positive margins. Ideally, all points should be classified correctly so that all the margins are positive. If all the points have been classified correctly and with the maximum possible certainty, the cumulative graph will be a single vertical line at $m = 1$. Figure 2 shows the margin cumulative distribution for the boosting classifier developed in this application. In this case, 8.96% of the negative margins match the training error. It should also be pointed out that about 40% of the observations have margins which are close to the unit (which shows those firms classified with a probability equal to one).

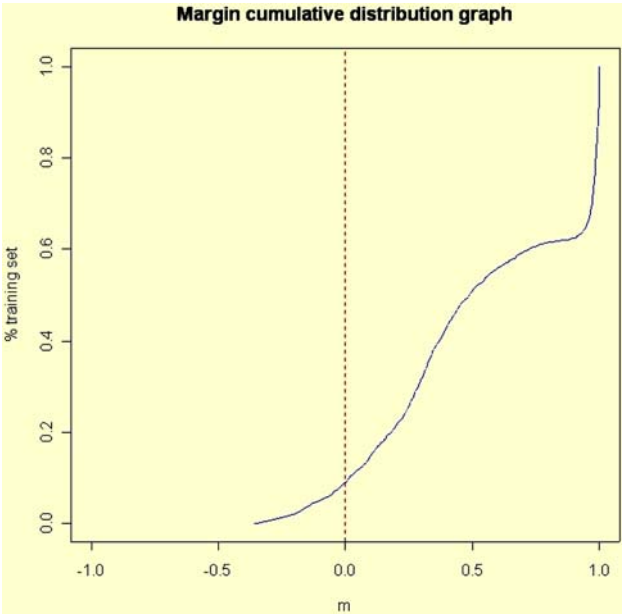

Fig. 2 Margin cumulative distribution graph

6 Conclusions

In this study, an ensemble classifier method has been analyzed, showing the improvement in accuracy that this method achieves both theoretically and empirically. As has been seen, boosting is based on building consecutive classifiers on modified versions of the training set which are generated according to the error rate of the previous classifier, while focusing on the hardest examples of the training set.

In the practical application, a wider concept of corporate failure have been used to include bankruptcy and temporary receivership firms as well as acquired and dissolved firms, and failed firms therefore include all four kinds of companies. The application has worked as usual with two classes, where healthy companies have been distinguished from failed ones, with the boosting method achieving a test error of 6.57%. Failed firms are therefore properly differentiated from healthy companies, and acquired firms are shown to behave in a more similar way to failed firms than healthy firms. This subject needs to be analyzed in the future.

Since the pioneering works of Beaver [16] and Altman [17], many studies have been developed to predict corporate

failure using accounting-based variables, and it does seem that there might be other quantitative and qualitative variables that can help prediction. In this research, the size of the firm, the activity sector and the legal structure have proved useful and the joint relative importance of these is 13.5%.

The most outstanding ratios (both for the individual classification tree and for boosting) are indebtedness and profitability, and these results are in line with previous studies of corporate failure.

Finally, it has been confirmed that the boosting method outperforms decision trees both in the cross-validation and test set estimation of the classification error, with the empirical comparison therefore demonstrating the superiority of ensemble methods over individual classification methods. The boosting method achieves better results than the pioneering studies listed in Appendix A with the exception of Wilson and Sharda's neural network (although, as already mentioned, caution should be exercised in this comparison).

This research has not addressed many important tasks such as the effect of the interdependence of combined classifiers on joint accuracy, the behavior of combination methods in the presence of noisy data, and the use of different basic classifiers for combinations. Consequently, these offer future lines of research.

Appendix A

Method	Study	Sample size	Number of predictors	Training error (%)	Test error (%)
Lineal discriminant analysis	Beaver [16]	158	6	10	No[1]
	Altman [17]	66	5	5	12
Logit	Ohlson [18]	2163	8	4	No
Classification Trees	Frydman et al. [4]	200	20	8	No
Neural Network	Wilson and Sharda [20]	129	5	0	2

[1] In these applications there is not a test set to estimate the error of the classifier.

Appendix B

Some of the following ratios are explained in the web http://faculty.philau.edu/lermackh/.

- JURIDICA. (Legal structure)
- NACE1 (NACE code at one digit)
- IN/AT1 (Net Incomes/Total Assets)
- BAI/AT1 (Earnings before interest and taxes/Total Assets)
- BAI/FP1 (Earnings before interest and taxes/Permanent Funds)
- AC/AT1 (Current Assets/Total Assets)
- T/AT1 (Cash/Total Assets)
- V/FP1 (Sales/Permanent Funds)
- AC/PC1 (Current Assets/Current liabilities)
- T/PC1 (Cash/Current liabilities)
- FM/V1 (Working Capital/Sales)
- PE/PT1 (Requirable Liabilities/Total Debt)
- CF/PT1 (Cash Flow/Total Debt)
- FM/AT1 (Working Capital/Total Assets)
- V/AT1 (Sales/Total Assets)
- V/AC1 (Sales /Current Assets)
- lnAT1 (Logarithm of Total Assets)
- lnAC1 (Logarithm of Current Assets)

References

1. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of the 13th international conference on machine learning. Morgan Kaufmann, pp 148–146.
2. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
3. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
4. Frydman H, Altman E, Kao D (1985) Introducing recursive partitioning for financial classification: the case of financial distress. J Finance 269–291
5. Friedman J, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 38(2):391–293
6. Banfield RE, Hall LO, Bowyer KW, Bhadoria D, Kegelmeyer WP, Eschrich S (2004) A comparison of ensemble creation techniques. In: Roli F, Kittler J, Windeatt T (eds) Multiple classifier systems, vol. 3077 of Lecture notes in computer science. Springer, Cagliari, Italy, pp 223–232
7. Dietterich TG (2000) Ensemble methods in machine learning. In: Kittler J, Roli F (eds) Multiple classifier systems, vol 1857 of Lecture notes in computer science. Springer, Cagliari, Italy, pp 1–15
8. Kuncheva LI, Bezdek JC, Duin RPW (2001) Decision templates for multiple classifier fusion: and experimental comparison. Pattern Recog 34:299–314
9. Lam L (2000) Classifier combinations: implementations and theoretical issues. In: Kittler J, Roli F (eds) Multiple classifier systems, vol 1857 of Lecture notes in computer science. Springer, Cagliari, Italy, pp 78–86
10. Valentini G, Masulli F (2002) Ensembles of learning machines. In: Marinaro M, Tagliaferri R (eds) Neural nets WIRN Vietri-02, vol. 2486 of Lecture notes in computer science. Springer-Verlag, Heidelberg, Germany, pp 3–19
11. Breiman L (1998) Arcing classifiers. Ann Statist 26(3):801–849
12. Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithm: bagging, boosting and variants. Mach Learn 36:105–142
13. Drucker H, Cortes C (1996) Boosting decision trees. In: Touretzky DS, Mozer MC, Hasselmo ME (eds) Advances in neural information processing systems 8: Proc. of the NIPS'95, vol. 8. The MIT Press, pp 479–485
14. Quinlan JR (1996) Bagging, boosting, and C4.5. In: Proceedings of the thirteenth national conference on artificial intelligence and the eighth innovative applications of artificial intelligence conference. AAAI Press/MIT Press, Menlo Park, pp 725–730

15. Schapire RE (2002) The boosting approach to machine learning: an overview. In: Workshop on nonlinear estimation and classification. MSRI
16. Beaver WH (1966) Financial ratios as predictors of failure. Empirical research in accounting. Selected studies. Supplement to vol. 4 of Journal of Accounting Research, pp 71–111
17. Altman EI (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J Finance 23(4):589–609
18. Olshon JA (1980) Financial ratios and the probabilistic prediction of bankruptcy. J Account Res 18(1):5–12
19. Zmijewski M (1984) Methodological issues related to the estimation of financial distress prediction models. J Account Res 22:59–86
20. Wilson RL, Sharda R (1994) Bankruptcy prediction using neural network. Decis Support Syst 11:545–557
21. Laitinen T, Kankaanpaa M (1999) Comparative analysis of failure prediction methods: the Finish case. Eur Account Rev 8(1): 67–92
22. R Development Core Team (2004) R: a language and environment for statistical computing. R foundation for statistical computing, Viena. http://www.R-project.org
23. Breiman L, Friedman JH, Olshen R, Stone CJ (1984) Clasification and regresion trees. Belmont, Wadsworth International Group
24. Opitz D, Maclin R (1999) Popular ensemble methods: an empirical study. J Artif Intell Res 11:169–198
25. Freund Y, Schapire RE, Bartlett P, Lee WS (1998) Boosting the margin: a new explanation for the effectiveness of voting methods. Ann Stat 26(5):1651–1686
26. Kuncheva LI (2004) Combining pattern classifiers. Methods and algorithms. Wiley

E. A. Cortés · M. G. Martínez · N. G. Rubio. The authors teach Statistics at the Faculty of Economic and Business Sciences in the University of Castilla-La Mancha. Esteban Alfaro completed his degree in Business in 1999 and got his Ph.D. in Economics in 2005, both in the University of Castilla-La Mancha. His thesis dealt with the application of ensemble classifiers to corporate failure prediction. Matías Gámez got his degree in Mathematics at the University of Granada in 1991 and finished a Master in Applied Statistics a year after. He completed his Ph.D. in Economics at the University of Castilla-La Mancha in 1998 on the application of geo-statistical techniques to the estimation of housing prices. Noelia García got her degree in Economics at the University of Madrid (UAM) in 1996 and completed her Ph.D. in Economics in 2004 on the construction of an intelligent and automated system for property valuation through the combination of neural nets and a geographic information system (GIS). Current research deals with spatial statistics and the combination of classifiers (decision trees and neural nets) for solving heated topics in the Economics.