

Bankruptcy forecasting: An empirical comparison of AdaBoost and neural networks[☆]

Esteban Alfaro^{a,*}, Noelia García^a, Matías Gámez^a, David Elizondo^b

^a *Economic and Business Sciences Faculty of Albacete, Castilla-La Mancha University, Plaza de la Universidad, 1. 02071 Albacete, Spain*

^b *School of Computing, De Montfort University, The Gateway, Leicester LE1 9BH, U.K.*

Received 21 February 2007; received in revised form 22 November 2007; accepted 3 December 2007

Available online 8 December 2007

Abstract

The goal of this study is to show an alternative method to corporate failure prediction. In the last decades Artificial Neural Networks have been widely used for this task. These models have the advantage of being able to detect non-linear relationships and show a good performance in presence of noisy information, as it usually happens, in corporate failure prediction problems. AdaBoost is a novel ensemble learning algorithm that constructs its base classifiers in sequence using different versions of the training data set. In this paper, we compare the prediction accuracy of both techniques on a set of European firms, considering the usual predicting variables such as financial ratios, as well as qualitative variables, such as firm size, activity and legal structure. We show that our approach decreases the generalization error by about thirty percent with respect to the error produced with a neural network.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Corporate Failure Prediction; Neural Network; AdaBoost

1. Introduction

Predicting corporate failure is a hot topic in management science due to its importance for making correct business decisions. The accuracy of the forecasting model is clearly of crucial importance in failure

prediction because many economic agents not only enterprises but financial institutions, auditors, consultants, policy makers or clients are affected by the bankruptcy of a firm. In classification terms, the type I error is especially important, i.e. when a firm which will fail in the future is classified as healthy. Owing to this fact many researchers have focused their effort on finding the most efficient classifier. In the last decades artificial neural networks have received special attention and several studies have dealt with failure forecasting using this technique. Here we present some of them only as examples. Wilson and Sharda [55] used a sample of 129 firms, 65 of which went bankrupt between 1975 and 1982 and 64 non bankrupt firms matched on

[☆] Work partially supported by the Spanish Government under grant TIN2006-07262 and by the Castilla-La Mancha University under grants TC20070075 and TC20070095.

* Corresponding author. Tel.: +34 967599200; fax: +34 967599220.

E-mail addresses: Esteban.Alfaro@uclm.es (E. Alfaro), Noelia.Garcia@uclm.es (N. García), Matias.Gamez@uclm.es (M. Gámez), elizondo@dmu.ac.uk (D. Elizondo).

industry and year. They applied resampling techniques to generate the training and test data sets and reached very satisfactory results with only five accounting ratios.

Serrano–Cinca [48] provided a data set made up of 66 Spanish banks, 29 of them in bankruptcy and the rest solvent. He used nine ratios chosen from amongst those most commonly employed in accounting empirical research. The author proved the superiority of the neural network model against linear discriminant analysis using the leaving one out estimation of the error. Charalambous et al [16] applied several neural networks methods to a dataset of 139 matched-pairs of bankrupt and non-bankrupt U.S. firms for the period 1983–1994. The authors compared the predictive performance of five methods, namely Learning Vector Quantization, Radial Basis Function, Feedforward networks that use the conjugate gradient optimization algorithm, the back-propagation algorithm and the logistic regression.

In this research, the neural network approach is compared to AdaBoosted [19] classification trees for predicting corporate failure. As far as we are aware, this is the first study to compare AdaBoost and Neural Networks capabilities for corporate failure prediction. To illustrate its usefulness, we will apply AdaBoost 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 to the United States, we will use financial ratios that have proved significant for predicting business failure in previous studies (e.g. Frydman [23]).

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 the model interpretation. This is clearly important in failure prediction as the firm must make appropriate decisions. Ensemble methods like AdaBoost do not improve model interpretation by themselves. Even more, they break the model interpretation conveyed by a decision tree. But, on the other hand, attribute importance methods can be devised to provide useful information for problem understanding. We will also calculate a novel 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. We use the legal definition of corporate failure which only includes bankrupt and temporary receivership firms. This is the

most common definition in corporate failure prediction literature. One numerical (the firm size) and two categorical variables (activity sector and legal structure) are included as descriptors in addition to the usual financial ratios. The AdaBoost method is applied to the failure prediction, analyzing the extent to which this methodology is suitable for the subject.

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

2. AdaBoost

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 [24]. 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. AdaBoost [19] 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 based on the training set.

AdaBoost applies the classification system repeatedly to the training data, but at each application, the learning attention is focused on different examples of this set using adaptive weights ($\omega_b(i)$), in contrast to other ensembles as Bagging [12] which do not update the weights. Once the training process has finished, the single classifiers obtained are combined into a final, highly accurate classifier based on 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 [5,8,13,17,22].

Even though there are several versions of the AdaBoost algorithm [22], the most widely used is the one by Freund and Schapire [19] which is known as AdaBoost. For simplification purposes, it can be assumed, without loss of generality, that there are only two classes. A training set is given by $T_n = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ where Y takes values of $\{-1, 1\}$. The weight $\omega_b(i)$ 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,

T^b , and is applied to each training sample. The error of this classifier is represented by ε_b and is calculated as

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

The new weight for the $(b+1)$ -th iteration will be

$$\omega_{b+1}(i) = \omega_b(i) \cdot \exp(\alpha_b \xi_b(i)). \quad (2)$$

where α_b is a constant calculated from the error of the classifier in the b -th iteration. More specifically, according to the authors mentioned above $\alpha_b = \ln((1 - \varepsilon_b) / \varepsilon_b)$.

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 (0.5). Therefore, the weights of the wrongly classified observations are increased and the weights of the correctly classified ones are decreased, forcing the single classifier built in the following iteration to focus on the hardest examples. Furthermore, the differences when the weights are updated are greater when the error of the single classifier is small since more importance 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 epoch. 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 $b=1, 2, 3, \dots, B$. Finally, the ensemble classifier is built as a linear combination of the single classifiers weighted by the corresponding constant α_b .

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

The AdaBoost algorithm is shown below:

AdaBoost algorithm (Freund and Schapire [19])

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

Freund and Schapire [20] showed that when the number B of iterations is increased, the training error level of the AdaBoost classifier exponentially tends to zero. They also demonstrated 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), the Vapnik-Chervonenkis's dimensionality coefficient of the parametric space of basic classifiers (d) and the number of iterations B in AdaBoost (number of combined single classifiers)

$$\hat{\varepsilon}_R = \hat{\varepsilon}_A + O \left(\sqrt{\frac{Bd}{n}} \right). \quad (4)$$

The generalization error of the final classifier may be reduced by increasing the size of the training data set. This error will increase when the number of single classifiers included increases. This is a sign of the classifier overfitting (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 being therefore inaccurate when classifying previously unseen samples).

3. Problem description

Predicting corporate failure is an important management science problem and its main goal is to differentiate 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.

Prediction of corporate failure is not new 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 [9] at a univariate level and Altman [2] with the application of multivariate analysis. Subsequently, and basically to overcome the restrictive statistical requirements of normality for the explanatory variables and equality for the variance-covariance group matrices, logit and probit models were also applied (Ohlson [36] and Zmijewski [58]). Classification trees or recursive partitioning proved useful in studies by Frydman et al.

[23]. More recently, artificial neural networks have been introduced as a powerful approach to this task (Odom and Sharda, [35]). For a more detailed study of neural network applications in bankruptcy forecasting see Chang Lee et al. [15], Laitinen and Kankaanpää [30], Perez [38] or the most recent work provided by Ravi Kumar and Ravi [44]. These papers present an interesting collection of the main failure prediction studies which are grouped according to the classification method used for prediction. Some of these results are shown in Table 1. Relating to the comparison of results, it should be noted that although classification accuracies of these studies are shown in Laitinen's work, care should be taken when comparing the prediction ability of alternative techniques in accounting because of the different starting conditions in alternative studies.

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. However and in order to facilitate comparisons with previous studies, we have used the most common corporate failure definition which uses a legal perspective and only includes bankruptcy and firms in temporary receivership as failed firms.

4. Data description

The companies used in this study 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.

Table 1
Results of previous experiences on bankruptcy forecasting [44]

Study	Sample size	Neural models ⁽¹⁾	Other methods	Best results with ...
Tam [51]	188	BP	LDA, F-L, K-NN, ID3	BP
Tam and Kiang [52]	202	FF, BP	LDA, LOGIT, K-NN, ID3	NN
Salchenberger [46]	862	BP	LOGIT	BP
Sharda and Wilson [49]	129	BP	LDA, LOGIT	BP
Fletcher and Goss [18]	36	BP	LOGIT	BP
Altman et al. [3]	3465	BP	LDA	LDA
Wilson and Sharda [55]	129	BP	LDA	BP
Tsukuda and Baba [53]	114	BP	LDA	BP
Lacher et al. [29]	282	CASCOR	LDA	CASCOR
Leshno and Spector [34]	88	P	LDA	P
Rahimian et al. [43]		BP, AT, P	LDA	BP
Serrano-Cinca [47]	129	SOFM with LDA, BP, P, AT		BP, P, AT
Serrano-Cinca [48]	66	FF	LDA	FF
Barniv et al [7]	237	BP	NPDA, LOGIT	BP
Bell [10]	2067	BP	LOGIT	BP
Piramuthu et al. [39]	182 202 48	BP, BPFC		BPFC
Kiviluoto [27]	1137	LVQ, SOFM, RBF-SOFM	K-NN	RBF-SOFM
Zhang et al. [57]	220	GRG2NN	LOGIT	GRG2NN
Yang et al. [56]	122	PNN-PN PNN, BP	FDA, LDA	PNN and BP with non-deflated data and FDA with deflated data
Charalambous [16]	278	SOFM, RBF, BP, LVQ FF (LSEF+CG)	LOGIT	FF (LSEF+CG)
Atiya [4]	1160	BP		BP with novel indicators
Kaski et al. [26]	1500	SOFM-E, SOFM-F		SOFM-F
Swicegood and Clark [50]	1741	BP	LDA, REG	BP
Back and Cho [6]	662	AANN, BP		AANN
Lam [31]	364	BP		BP integrating fundamental and technical analysis
Lee et al. [32]	166	LDA-ASBP ID3-ASBP SOFM-ASBP	LDA, ID3	Hybrid BP models
Lee et al. [33]	168	BP, SOFM	LDA	BP

⁽¹⁾AANN=Auto associative neural networks, AT=Athena, BP=Backpropagation, CASCOR=Cascade correlation neural network, CG=Conjugate Gradient, FF=Feedforward, F-L=Factor-logistic, GRG2NN=Generalized reducing gradient, ID3-ASBP=ID3 assisted Backpropagation, K-NN=K-nearest neighbour, LDA=Linear Discriminant Analysis, LDA-ASBP=LDA assisted Backpropagation, LVQ=Learning Vector Quantization, MLP=Multilayer Perceptron, P=Single layer Perceptron, RBF=Radial Basis Functions, REG=Regulators, SOFM=Self Organizing Feature Maps, SOFM-ASBP=SOFM assisted Backpropagation, SOFM-E=Euclidean Self Organizing Maps, SOFM-F=Fisher metric Self Organizing Maps.

In the case of failed firms, firms which had failed (bankruptcy and temporary receivership) 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 from 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 the 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, 590 firms were randomly selected for each group (failed/healthy), obtaining 1180 observations for the total data set. Instead of pairing the failed/healthy firms by sector or size as several failure studies have done, these 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 explanatory variable with three options: public corporations, limited corporations, and other corporations.

In addition, thirteen 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. Sixteen 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 Table 2.

Once the explanatory variables were selected, we carried out an exploratory analysis of the quantitative data. Tables 3 and 4 show some statistics, normality tests for each variable and the correlation matrix. From these results it is worth to point out that none of the variables follows a normal distribution and that there is a high degree of correlation among the variables.

Table 2
Description of variables

Variable	Description	Variable	Description
CA.TA	Current Assets/Total Assets	L.TD	Liabilities/Total Debt
CA.CL	Current Assets/Current Liabilities	C.TA	Cash/Total Assets
EBIT.TA	Earnings before interest and taxes/Total Assets	C.CL	Cash/Current liabilities
CF.TD	Cash Flow/Total Debt	S.CAP	Sales/Capital
NACE1	NACE code at one digit	EBIT.CAP	Earnings before taxes/Capital
WC.TA	Working Capital/Total Assets	lnTA	Logarithm of Total Assets
WC.S	Working Capital/Sales	S.CA	Sales /Current Assets
LE	Legal structure	S.TA	Sales /Total Assets

Some of the following ratios are explained in White [54] and also in the web <http://faculty.philau.edu/lermackh/>.

Furthermore, Table 5 shows the discriminant capacity of each variable and the misclassification percentage of the univariate approach (LDA). Although some variables present acceptable discriminant power, we decided not to apply linear techniques based on the Fisher approach (linear discriminant or logistic regression) because the conditions to be optimal could not be satisfied (normality and unitary covariance matrix). As an alternative to overcome this problem, we could use natural logarithm transformations to approximate a normal distribution even though the transformed variables may be very difficult to interpret in this area. However, as it will be shown in Section 5, we applied the LDA in order to have a basic reference to be able to compare the results provided by the Adaboost and Artificial Neural Network model.

5. Experimental results

In this paper, the same failure prediction problem is solved using two different classification methods in order to compare their classification accuracies in this task. To estimate the real accuracy, the total initial sample of 1180 Spanish companies was divided into two sets: eighty percent were used as a training set to build the classifier, and the rest were hidden from the classification method and were presented as new data to check the prediction accuracy. The training set therefore comprised 472 healthy firms and a further 472 failed firms (944 firms represent 80% of the total set). The test set consisted of 236 firms, with an equal number of healthy and failed firms (20% of the total). As mentioned before, 16 diagnostic variables were used for

Table 3
Results obtained from exploratory data analysis

	EBIT.TA	EBIT.CAP	CA.TA	C.TA	S.CAP	CA.CL	C.CL	WC.S	L.TD	CF.TD	WC.TA	S.TA	S.CA	lnTA
Mean	1.5514	0.5328	0.5197	0.1474	8.1580	2.7673	0.7119	2.0219	−0.3856	0.8728	0.0187	1.0233	1.0065	7.5469
Median	0.0784	0.3855	0.5984	0.0404	0.7804	1.1235	0.1025	0.0670	0.6832	0.0624	0.0620	0.8442	1.4837	7.4184
Std. Dev.	54.8164	4.3662	5.7313	1.8576	63.1374	17.1474	18.0535	17.2610	10.0915	10.1956	4.9913	1.3517	44.4035	1.5616
Asymmetry	31.6723	14.3114	−24.9542	20.3904	2.8566	−2.4655	−20.7833	0.6569	24.3215	−27.2205	−24.2789	−1.5633	−33.6943	0.5617
Skewness	1072.4834	305.8308	846.8311	733.5410	216.9099	182.5282	604.3463	95.7141	757.4138	872.4553	782.4589	36.3493	1147.4768	1.9395
Minimum	−394.3403	−25.1707	−179.9724	−27.4740	−1106.5820	−294.5847	−521.8325	−240.3293	−46.8859	−324.0337	−154.2119	−16.2484	−1514.6670	1.4081
Maximum	1840.4110	102.4375	66.5612	56.0455	1225.0620	283.4821	121.3837	239.3921	310.0544	60.1981	49.0374	10.4067	66.2963	15.0009
1st Quartil	0.0194	0.1390	0.1790	0.0117	0.0468	0.8083	0.0200	−0.0786	−1.1317	0.0114	−0.0474	0.0577	0.4463	6.5744
3rd Quartil	0.2778	0.6984	0.8683	0.1022	8.1562	2.0993	0.5633	0.7276	0.9118	1.5821	0.2064	1.6190	2.5122	8.4297
SW Test	0.0180	0.1812	0.0493	0.0563	0.2205	0.2405	0.1053	0.2706	0.1711	0.1228	0.0599	0.7327	0.0275	0.9745
SW <i>p</i> -value	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
KS test	0.3743	0.3558	0.4992	0.4992	0.4418	0.5559	0.4559	0.2465	0.2792	0.3651	0.3122	0.4738	0.4858	0.9970
KS <i>p</i> -value	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4
Correlation matrix

	EBIT.TA	EBIT.CAP	CA.TA	C.TA	S.CAP	CA.CL	C.CL	WC.S	L.TD	CF.TD	WC.TA	S.TA	S.CA	lnTA
EBIT.TA	1.0000	0.0356	0.5241	0.9500	−0.0035	0.0019	0.0031	−0.0092	−0.0121	0.1012	0.4728	−0.2322	−0.0009	−0.0972
EBIT.CAP	0.0356	1.0000	0.0122	0.0309	0.2833	0.0011	0.0019	0.0115	0.0007	0.0105	0.0119	0.0064	−0.0025	−0.0019
CA.TA	0.5241	0.0122	1.0000	0.6997	0.0067	0.0084	−0.0007	0.0124	0.0015	0.0290	0.9603	0.2638	0.0020	0.0107
C.TA	0.9500	0.0309	0.6997	1.0000	−0.0061	−0.0035	−0.0368	−0.0290	−0.0252	0.0950	0.6282	−0.1172	0.0015	−0.0936
S.CAP	−0.0035	0.2833	0.0067	−0.0061	1.0000	−0.0119	−0.0044	−0.0152	0.0160	−0.0106	−0.0008	0.1428	0.0056	0.0366
CA.CL	0.0019	0.0011	0.0084	−0.0035	−0.0119	1.0000	0.2627	0.1035	−0.0589	0.0357	0.0197	−0.0684	−0.0010	−0.0223
C.CL	0.0031	0.0019	−0.0007	−0.0368	−0.0044	0.2627	1.0000	0.0222	0.0936	−0.0442	0.0015	−0.0229	0.0019	0.0131
WC.S	−0.0092	0.0115	0.0124	−0.0290	−0.0152	0.1035	0.0222	1.0000	−0.0669	0.0612	0.0339	−0.0879	0.0062	−0.0104
L.TD	−0.0121	0.0007	0.0015	−0.0252	0.0160	−0.0589	0.0936	−0.0669	1.0000	−0.9526	0.0025	0.0993	0.0625	−0.0505
CF.TD	0.1012	0.0105	0.0290	0.0950	−0.0106	0.0357	−0.0442	0.0612	−0.9526	1.0000	0.0322	−0.1002	−0.0552	0.0183
WC.TA	0.4728	0.0119	0.9603	0.6282	−0.0008	0.0197	0.0015	0.0339	0.0025	0.0322	1.0000	0.2098	−0.0019	0.0369
S.TA	−0.2322	0.0064	0.2638	−0.1172	0.1428	−0.0684	−0.0229	−0.0879	0.0993	−0.1002	0.2098	1.0000	0.0435	−0.1790
S.CA	−0.0009	−0.0025	0.0020	0.0015	0.0056	−0.0010	0.0019	0.0062	0.0625	−0.0552	−0.0019	0.0435	1.0000	−0.0401
lnTA	−0.0972	−0.0019	0.0107	−0.0936	0.0366	−0.0223	0.0131	−0.0104	−0.0505	0.0183	0.0369	−0.1790	−0.0401	1.0000

Table 5

ANOVA and discriminatory capacity for each variable

	EBIT. TA	EBIT. CAP	CA. TA	C.TA	S.CAP	CA.CL	C.CL	WC.S	L.TD	CF.TD	WC. TA	S.TA	S.CA	lnTA
ANOVA for each variable														
<i>F</i> value	0.9811	0.0486	1.7694	2.7768	12.4850	10.3950	1.3542	19.339	20.226	9.4274	0.1121	294.38	1.0849	0.0675
Prob ($>F$)	0.3221	0.8256	0.1837	0.0959	0.0004	0.0013	0.2448	0.0000	0.0000	0.0022	0.7378	0.0000	0.2978	0.7951
LDA using each variable as explanatory (50 experiments: 5×10 folders cross validation)														
Mean error rate	0.4961	0.5097	0.4331	0.4654	0.3383	0.3595	0.4537	0.3304	0.2138	0.2705	0.4878	0.2214	0.4996	0.4966

each company with information for the year prior to the moment of failure.

In order to apply the AdaBoost algorithm, we have used the *adabag* library [1] for the R statistical program. This program consists of a series of packages for data manipulation, calculus, and graphics [42]. 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 the difference that the R program is a freely available software program which can be downloaded from <http://cran.r-project.org/>. 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. On the other hand, the neural networks models have been implemented using the TRAJAN software (<http://www.trajan-software.demon.co.uk/>).

5.1. Corporate failure prediction using neural networks

An artificial neural network is an information processing device inspired on biological nervous systems. Generally these models consist of a set of computational units, also called neurons, organized in layers. Each neuron is connected to other neurons through synaptic junctions called synapses. The learning process is shown through changes in strength connections. There are several models of artificial neural networks. The most common architecture is the Multilayer Perceptron [11,25], a feedforward network that consists of an input layer, an output layer and a number of hidden layers. This kind of nets is fundamentally characterized by the fact that information signal is propagated from the input nodes until it has reached an output node without cycles of links. The goal is to find suitable values for the synaptic weights so that minimize the deviation between the output given by the net and the desired output. This process is known as supervised learning and the most known mechanism for weight adaptation is the back-

propagation algorithm [45]. The B-P method uses gradient descent to change the weights proportional to the derivative of the error function with respect to each weight. The procedure is repeated for all patterns in the training set. The patterns are continuously presented and weights are adjusted until the error is sufficiently low.

The neural network model selected in this work is the well-known Multilayer Perceptron with one hidden layer. The number of nodes in the input and the output layers has been set by the structure of our analysis, i. e. the number of explanatory and output variables, respectively. On the other hand, several experiments were conducted to find the number of hidden elements which gave the greatest accuracy in predicting the test data set. Table 6 shows the results (minimum, maximum and mean percentage of correctly classified cases) obtained from twenty different topologies,

Table 6

Impact of the number of hidden nodes in the percentage of correctly classified cases (10 experiments for each topology)

# hidden layers	% of correctly classified cases		
	Min	Max	Mean
1	57.63	58.47	58.39
2	53.39	80.51	61.86
3	57.63	71.19	60.68
4	55.08	70.34	59.49
5	55.93	66.95	59.32
6	56.78	63.56	59.24
7	56.78	61.86	58.64
8	57.63	76.27	61.78
9	57.63	66.95	60.00
10	56.78	60.17	58.47
11	56.78	64.41	59.32
12	55.93	84.75	64.49
13	55.93	77.97	60.51
14	57.63	77.12	62.88
15	57.63	75.42	60.51
16	56.78	72.03	60.42
17	56.78	77.12	63.31
18	56.78	67.80	59.92
19	57.63	82.20	63.05
20	56.78	69.49	61.19

varying the number of hidden elements from 1 to 20 and training ten times each topology. This procedure resulted in the selection of a 16:27-12-1:1 network, i.e. an input layer with sixteen nodes, pre-processed into twenty-seven nodes (The qualitative variables LE and NACE1 have been encoded using the one-of-N method. This technique uses a set of dichotomous variables, one for each possible nominal value. Three nodes have been necessary for LE and ten for NACE1. So, the total number of input variables changed from 16 to 27), a hidden layer with twelve elements, and finally an output layer with one. This architecture is shown in Fig. 1.

In order to train the network, the following decisions were set. Firstly, the above mentioned division of the sample cases into two subsets: training and test. The activation functions were selected to be linear in

the input layer and sigmoid in the hidden and output layers. The error function selected was the sum of squared errors and the learning algorithm to minimize it was Back Propagation with adaptive learning starting at 0.3 and finishing at 0.01 and momentum term set at the value 0.3. After 1868 epochs the network was trained and the following results were yielded. The main result from the confusion matrix in Table 7 is the error percentage, measured as the percentage of wrongly classified cases. The error is closed to 11% for the training set and it is a little higher for the test set (12.712%). It is worth stressing that if we consider the most important error as the classification of a failed firm as a healthy one (Type I error), the behaviour of the network is very satisfactory since it is about 4.025% in the training set (7.627% in the test set).

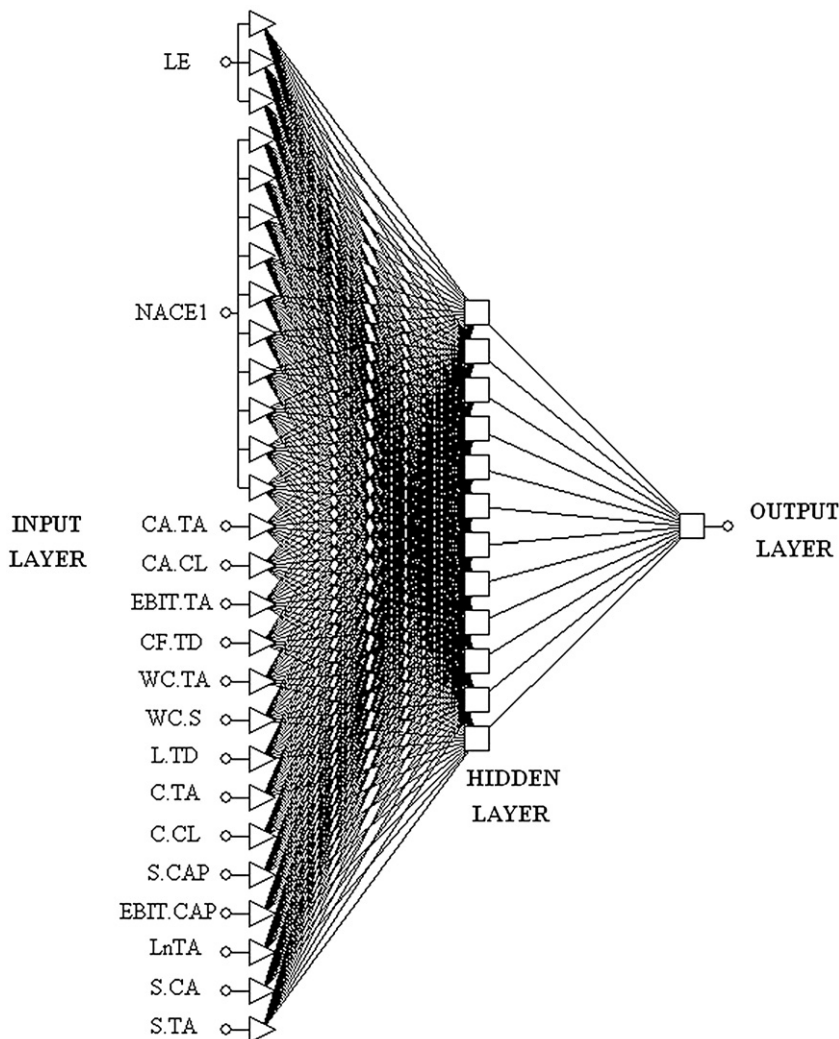


Fig. 1. MLP 16:27-12-1:1 architecture.

Table 7
Confusion matrices and errors of the Neural Network classifier

Predicted class	Observed class				
	N. Network (MLP)	Training	10.805%	Test	12.712%
		Failed	Healthy	Failed	Healthy
Failed		453	83	109	21
Healthy		19	389	9	97

The relative contribution of each input variable to the global performance can be assessed by means of a sensitivity analysis, which entails testing the performance of the network as if the input variables were unavailable. This ratio measures the relation of the error if the correspondent input variable is unavailable with the error if all the variables are available. A ratio of one or lower therefore means that pruning this input variable has no effect on the performance of the network. Table 8 shows the ranking of the input variables with a ratio above 1.05, which means that these variables can be considered as the most important ones in terms of their discriminatory power. According to the sensitive analysis results, the most important ratios are the efficiency, the sector the proxy variable for the firm size, the legal structure and the level of indebtedness. The first ratio refers to as the rate of sales over the total assets and shows, in a figurative sense, how many times the asset has been sold and, therefore, its number of turnovers, i.e. how many times it has been sold and replaced. This ratio can also be interpreted as the amount of sales in euros by each euro invested in assets, reflecting the capacity of assets to generate sales and the firm relative efficiency on managing them. The higher this ratio, for the same commercial margin, the higher the profit for a lower invest and, therefore, the higher the profitability.

It is worth to stress that the following variables in the ranking are not the usual financial ratios but the additional firm information proposed as potentially useful in this research.

The fifth ratio is L.TD (Liabilities/Total Debt). This ratio considers the weight of 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.

5.2. Corporate failure prediction using AdaBoost

AdaBoost can use any sort of classification system as the individual classifier, but we have used decision trees [14] in this application basically for the following three reasons: decision trees are used in most AdaBoost applications, they achieve good results and classification trees easily handle qualitative variables. Very briefly, we can see a classification tree like a method to express knowledge and aid decision making. This technique provides a way to encapsulate and structure the knowledge of experts to be used by less-experienced user. The construction of classification trees aims to find a decision tree that is as small as possible and fits the data. This way it is assumed that it is more likely to find a good generalization of the training data. A tree is first built as a given set of pre-classified training examples partitioned recursively according to a chosen attribute at each node. The tree-building process is further divided into three steps: attributes evaluation, splitting point selection and training data partition. Then a bottom-up tree pruning process follows to remove branches with lower predictive power, once the whole decision tree has been built. Pruning techniques, such as those used in the most known algorithms ID3 [40], CART [14] and C4.5 [41] have proved to be really useful in order to avoid overfitting.

The adabag package has three functions dealing with AdaBoost 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 is stored or the name of the variable that contains the observation class and the explanatory variables, the number of individual trees to be used, and the size of these trees.

Table 8
Sensitive analysis

Ranking	Variable	Ratio
1	S.TA	1.3091
2	NACE1	1.2109
3	LnTA	1.1585
4	LE	1.1311
5	L.TD	1.0964

Table 9
Confusion matrices and errors of the AdaBoost classifier

Predicted class	Observed class				
	AdaBoost	Training	7.627%	Test	8.898%
		Failed	Healthy	Failed	Healthy
Failed		460	60	114	17
Healthy		12	412	4	101

Table 10
Comparison of results with other methods

Models	Error rates					
	Training			Testing		
	Overall	Type I	Type II	Overall	Type I	Type II
LDA	20.763%	24.364%	17.161%	20.339%	28.814%	11.864%
C.Trees	10.452%	0.753%	20.155%	12.714%	3.390%	22.034%
ANN	10.805%	4.025%	17.585%	12.712%	7.627%	17.797%
AdaBoost	7.627%	2.523%	12.712%	8.898%	3.390%	14.407%

Using the functions described above an AdaBoost classifier is built with 100 trees that have been pruned using a maximum depth of 2 to limit the size of the individual tree in each AdaBoost epoch. The depth of the tree is the distance between a leaf node and the root node. Since the test error is of 8.898%, there is a reduction of 30% compared with the neural network test error, which is of 12.712%. 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 while the Type I error is 3.39% in the test set. Table 9 shows the errors of the AdaBoost classifier in the training and test sets and their confusion matrices.

As we mentioned in the previous section a LDA was carried out only with the purpose of having a reference point. As it can be seen in Table 10, the results from the linear model are much less satisfactory with a percentage of misclassification over 20% in both (training and test) data sets. Table 10 also shows the results provided by a single tree pruned by the rule “minimum error in cross-validation plus one standard deviation”. These results are very similar to the neural network ones.

In order to ensure that the comparison between the neural network model and the AdaBoost ensemble does not happen by chance, we used five repetitions of 10-fold cross-validation (Opitz and Maclin [37]). The entire set (1180 firms) is used for each of the 10-fold cross-validation experiment. This way we obtained the error rates for the AdaBoost and the Neural Networks on each one of the 50 experiments. Once we checked the normality of the error distributions of both classification methods using the tests shown in Table 11, we can apply

the test for comparing the means of two normal distributions. To achieve a higher degree of certainty we used the one tail test, establishing the null hypothesis that the average error of the AdaBoost classifier is equal or higher than the neural network average error against the alternative hypothesis that it is lower than the neural network one. This way, if we reject the null hypothesis, we will be almost sure that the alternative one is true. The result obtained is enlightening because the *t* statistic is -8.0884 and its corresponding *p*-value is $8.821 \bullet 10^{-13}$. So we can reject the null hypothesis and state that the differences are statistically significant with AdaBoost ahead. In fact, AdaBoost reduces the average error in the cross-validation analysis by 28.04% compared with the neural network.

The AdaBoost function of the adabag package allows us to quantify the relative importance of the predictor variables. This is a really important advantage because it is difficult to interpret the hundreds or thousands of trees used in the AdaBoost 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 12 shows all variables arranged from the greatest to least relative importance. In this case, the most important ratios are EBIT.TA, L.TD, S.TA and LnTA with values at this measure of 17.33, 13.72, 10.47 and 9.02%, respectively. Those variables which are different from financial ratios (NACE1, LnTA and LE) have an interesting contribution of 20.2% in total.

Table 11
Normality tests, average and standard deviation error

Test	N.Network	AdaBoost
Kolmogorov–Smirnov	0.1454	0.1103
KS. <i>p</i> value	0.2408	0.5768
Shapiro	0.9559	0.9766
Shapiro <i>p</i> value	0.0602	0.4196
Average error	0.1359	0.0978
Standard deviation	0.0249	0.0222

Table 12
Relative importance of variables

Variable	Relative importance	Variable	Relative importance
EBIT.TA	17.33	CA.TA	5.05
L.TD	13.72	EBIT.CAP	3.61
S.TA	10.47	LE	2.88
LnTA	9.02	S.CAP	2.17
C.CL	8.66	WC.TA	2.17
NACE1	8.30	CA.CL	1.80
CF.TD	6.50	S.CA	1.80
C.TA	5.05	WC.S	1.44

The most important ratios in the analysis are the economic profitability, level of indebtedness, efficiency, and the proxy variable for the firm size. The first of these ratios shows the corporation's success in the application of assets, measuring them by means of the weight of generated earnings before interest and taxes on this accounting magnitude. The most efficient firms in this aspect will undoubtedly have a greater likelihood of being classified as healthy. The variables in the positions from second to fourth in this ranking appeared also in the sensitive analysis of the neural network, although they are placed in different order.

In the AdaBoost literature, the concept of margin [21] 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 $\mu_j(x)$, $j=1,2,\dots,q$ as

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

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 [28] 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 correctly classified, 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 correctly classified and with the maximum possible certainty, the cumulative graph will be a single vertical line at $m=1$. Fig. 2 shows the margin cumulative distribution for the AdaBoost classifier developed in this application. In this case, 7.63% of the negative margins match the training error. It should also be pointed out that about 25% of the observations have margins which are close to the unit (which shows those firms that have been classified with a probability equal to one).

6. Conclusions

In this study, two classification methods have been compared, showing the improvement in accuracy that AdaBoost achieves against the Neural Network. As has been seen, AdaBoost 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, the legal concept of corporate failure have been used which includes bankruptcy and temporary receivership firms. The application has worked as usual with two classes, where healthy companies have been distinguished from failed ones, with the AdaBoost method achieving a test error of 8.898%. Failed firms are therefore properly differentiated from healthy companies. This result means that the AdaBoost strategy for combining single trees achieves a reduction of 30% in the test error compared with the individual neural network.

Moreover, it has been confirmed that the AdaBoost ensemble of trees outperforms Neural Networks both in the cross-validation and test set estimation of the classification error, with the empirical comparison therefore, demonstrating the superiority of simple trees ensembles over individual neural networks.

Since the pioneering works of Beaver [9] and Altman [2], 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 20.2%.

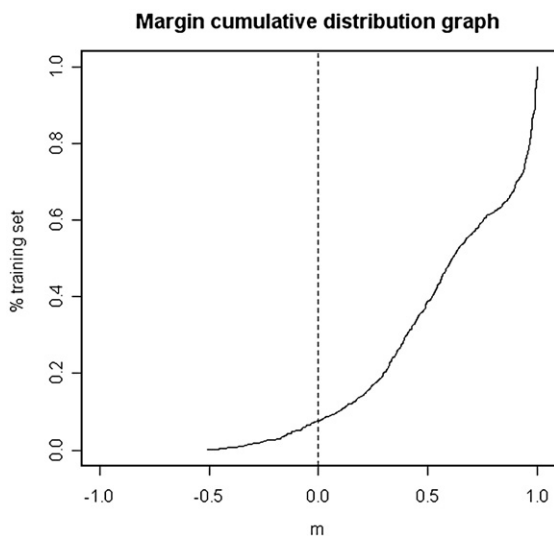


Fig. 2. Margin cumulative distribution graph.

The most outstanding ratios for AdaBoost are the economic profitability, level of indebtedness, efficiency, and the proxy variable for the firm size, and these results are in line with previous studies of corporate failure.

This research has not addressed many important tasks such as the effect of the interdependence of combined classifiers on joint accuracy or the behaviour of combination methods in the presence of noisy data. Our immediate task is the use of neural networks as basic classifiers for AdaBoost. Consequently, these offer future lines of research.

References

- [1] E. Alfaro, M. Gámez, N. García, Adabag: implements AdaBoost. M1 and bagging. R package version 1.0, <http://www.R-project.org> 2006.
- [2] E.I. Altman, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, *Journal of Finance* 23 (4) (1968) 589–609.
- [3] E.I. Altman, G. Marco, F. Varetto, Corporate distress diagnosis: comparison using linear discriminant analysis and neural networks (the Italian experience), *Journal of Banking and Finance* 18 (1994) 505–529.
- [4] A.F. Atiya, Bankruptcy prediction for credit risk using neural networks: a survey and new results, *IEEE Transaction on Neural Networks* 12 (4) (2001) 929–935.
- [5] R.E. Banfield, L.O. Hall, K.W. Bowyer, D. Bhadoria, W.P. Kegelmeyer, S. Eschrich, A comparison of ensemble creation techniques, in: F. Roli, J. Kittler, T. Windeatt (Eds.), *Multiple Classifier Systems*, Vol. 3077 of Lecture Notes in Computer Science, Springer, Cagliari, 2004, pp. 223–232.
- [6] J. Baek, S. Cho, Bankruptcy prediction for credit risk using an auto associative neural network in Korean firms, *Proceeding of the International Conference on Computational Intelligence for Financial Engineering*, IEEE Press, Hong Kong, 2003, pp. 25–29.
- [7] R. Barniv, A. Anuragh, R. Leach, Predicting the out come following bankruptcy filing: a three state classification using NN, *International Journal of Intelligence Systems in Accounting, Finance and Management* 6 (1997) 177–194.
- [8] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithm: bagging, boosting and variants, *Machine Learning* 36 (1999) 105–142.
- [9] W.H. Beaver, Financial ratios as predictors of failure, *Empirical Research in Accounting. Selected Studies. Supplement to Vol. 4 of Journal of Accounting Research*, Blackwell Publishing, 1966, pp. 71–111.
- [10] T.B. Bell, Neural nets or the logit model? A comparison of each model's ability to predict commercial bank failures, In *International Journal of Intelligence Systems in Accounting, Finance and Management* 6 (1997) 249–264.
- [11] C.M. Bishop, *Neural networks for pattern recognition*, Clarendon Press, Oxford, 1995.
- [12] L. Breiman, Bagging predictors, *Machine Learning* 24 (2) (1996) 123–140.
- [13] L. Breiman, Arcing classifiers, *The Annals of Statistics* 26 (3) (1998) 801–849.
- [14] L. Breiman, J.H. Friedman, R. Olshen, C.J. Stone, *Classification and regression trees*, Wadsworth International Group, Belmont, 1984.
- [15] K. Chang Lee, I. Han, Y. Kwon, Hybrid neural network models for bankruptcy predictions, *Decision Support Systems* 18 (1) (1996) 63–72.
- [16] C. Charalambous, A. Charitou, F. Kaorou, Comparative analysis of artificial neural network models: application in bankruptcy prediction, *Annals of Operation Research* 99 (4) (2000) 403–425.
- [17] T.G. Dietterich, Ensemble methods in machine learning, in: J. Kittler, F. Roli (Eds.), *Multiple Classifier Systems*, Vol. 1857 of Lecture Notes in Computer Science, Springer, Cagliari, 2000, pp. 1–15.
- [18] D. Fletcher, E. Goss, Application forecasting with neural networks. An application using bankruptcy data, *Information and Management* 24 (1993) 159–167.
- [19] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, *Proc. 13th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, 1996, pp. 148–156.
- [20] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* 55 (1) (1997) 119–139.
- [21] Y. Freund, R.E. Schapire, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, *The Annals of Statistics* 26 (5) (1998) 1651–1686.
- [22] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, *The Annals of Statistics* 38 (2) (2000) 391–393.
- [23] H. Frydman, E. Altman, D. Kao, Introducing recursive partitioning for financial classification: the case of financial distress, *Journal of Finance* (1985) 269–291.
- [24] D.J. Hand, *Discrimination and classification*, John Wiley, New Jersey, 1981.
- [25] S. Haykin, *Neural networks. A comprehensive foundation*, Prentice Hall, New York, 1994.
- [26] S. Kaski, J. Sinkkonen, J. Peltonen, Bankruptcy analysis with self-organizing maps in learning metrics, *IEEE Transaction on Neural Networks* 12 (4) (2001).
- [27] K. Kiviluoto, Predicting bankruptcies with self organizing maps, *Neurocomputing* 21 (1998) 191–201.
- [28] L.I. Kuncheva, *Combining pattern classifiers. Methods and algorithms*, Wiley, New Jersey, 2004.
- [29] R.C. Lacher, P.K. Coats, S.C. Sharma, L.F. Fante, A neural network for classifying the financial health of a firm, *European Journal of Operational Research* 85 (1995) 53–65.
- [30] T. Laitinen, M. Kankaanpää, Comparative analysis of failure prediction methods: the Finnish case, *European Accounting Review* 8 (1) (1999) 67–92.
- [31] M. Lam, Neural network techniques for financial performance prediction: integrating fundamental and technical analysis, *Decision Support Systems* 37 (2004) 567–581.
- [32] K.C. Lee, I. Han, Y. Kwon, Hybrid neural network models for bankruptcy predictions, *Decision Support Systems* 18 (1996) 63–72.
- [33] K. Lee, D. Booth, P. Alam, A comparison of supervised and unsupervised neural networks in predicting bankruptcy of Korean firms, *Expert Systems with Applications* 29 (2005) 1–16.
- [34] M. Leshno, Y. Spector, Neural network prediction analysis: the bankruptcy case, *Neurocomputing* 10 (1996) 125–147.
- [35] M. Odom, R. Sharda, A neural network for bankruptcy prediction, *Proceedings of the International Joint Conference on Neural Networks*, IEEE Press, San Diego, CA, 1990.
- [36] J.A. Olshon, Financial ratios and the probabilistic prediction of bankruptcy, *Journal of Accounting Research* 18 (1) (1980) 5–12.
- [37] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, *Journal of Artificial Intelligence Research* 11 (1999) 169–198.

- [38] M. Pérez, Artificial neural networks and bankruptcy forecasting: a state of the art, *Neural Computing & Applications* 15 (2) (2006) 154–163.
- [39] S. Piramuthu, H. Ragavan, M.J. Shaw, Using feature construction to improve the performance of the neural networks, *Management Sciences* 44 (3) (1998).
- [40] J.R. Quinlan, Induction on decision trees, *Mach. Learn.* 1 (1986) 81–106.
- [41] J.R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, 1993.
- [42] R Development Core Team, R: a language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, <http://www.R-project.org> 2004.
- [43] E. Rahimian, S. Singh, T. Thammachote, R. Virmani, Bankruptcy prediction by neural networks, in: R.R. Trippi, E. Turban (Eds.), *Neural Networks in Finance and Investing*, Irwin Professional Pub., Burr Ridge, 1996.
- [44] P. Ravi Kumar, V. Ravi, Bankruptcy prediction in banks and firms via statistical and intelligent techniques — a review, *European Journal of Operational Research* 180 (2007) 1–28.
- [45] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representation by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, The MIT Press, Cambridge, 1986, pp. 318–362.
- [46] L. Salchenberger, C. Mine, N. Lash, Neural networks: a tool for predicting thrift failures, *Decision Sciences* 23 (1992) 899–916.
- [47] C. Serrano-Cinca, Self organizing neural networks for financial diagnosis, *Decision Support Systems* 17 (3) (1996) 227–238.
- [48] C. Serrano-Cinca, Feedforward neural networks in the classification of financial information, *European Journal of Finance* 3 (3) (1997) 183–202.
- [49] R. Sharda, R.L. Wilson, Performance comparison issues in neural network experiments for classification problems, *Proceedings of the 26th Hawaii International Conference on Systems Sciences*, IEEE Press, Hawaii, 1993.
- [50] P. Swicegood, J.A. Clark, Off-site monitoring for predicting bank under performance: a comparison of neural networks, discriminant analysis and professional human judgement, *International Journal in Accounting, Finance and Management* 10 (2001) 169–186.
- [51] K.Y. Tam, Neural networks models and the prediction of bank bankruptcy, *Omega* 19 (5) (1991) 429–445.
- [52] K.Y. Tam, M. Kiang, Predicting bank failures: a neural network approach, *Decision Sciences* 23 (1992) 926–947.
- [53] J. Tsukuda, S.I. Baba, Predicting Japanese corporate bankruptcy in terms of finance data using neural network, *Computers and Industrial Engineering* 27 (1–4) (1994) 445–448.
- [54] G.I. White, A.C. Sondhi, D. Fried, *The analysis and use of financial statements*, John Wiley & Sons, Inc., 2003.
- [55] R.L. Wilson, R. Sharda, Bankruptcy prediction using neural network, *Decision Support Systems* 11 (1994) 545–557.
- [56] Z.R. Yang, M.B. Platt, H.D. Platt, Probability neural networks in bankruptcy prediction, *Journal of Business Research* 44 (1999) 67–74.
- [57] G. Zhang, M.Y. Hu, B.E. Patuwo, D.C. Indro, Artificial neural networks and bankruptcy prediction general framework and cross-validated analysis, *European Journal of Operational Research* 116 (1999) 16–32.
- [58] M. Zmijewski, Methodological issues related to the estimation of financial distress prediction models, *Journal of Accounting Research* 22 (1984) 59–86.



Esteban Alfaro Cortés teaches Statistics at the Faculty of Economic and Business Sciences in the University of Castilla-La Mancha. He 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. Current research deals with spatial statistics and the combination of classifiers (decision trees and neural nets) for solving heated topics in the Economics.



Noelia García Rubio teaches Statistics at the Faculty of Economic and Business Sciences in the University of Castilla-La Mancha. She 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.



Matías Gámez Martínez teaches Statistics at the Faculty of Economic and Business Sciences in the University of Castilla-La Mancha. He 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. Current research deals with spatial statistics and the combination of classifiers (decision trees and neural nets) for solving heated topics in the Economics.



D. Elizondo received the B.Sc. degree in computer science from Knox College, Galesburg, IL, in 1986, the M.Sc. degree in artificial intelligence from the University of Georgia, Athens, in 1992, and the Ph.D. degree in computer science from the Université Louis Pasteur, Strasbourg, France, and the Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP), Martigny, Switzerland, in 1996. He is currently a Senior Lecturer at the Centre for Computational Intelligence of the School of Computing at De Montfort University, Leicester, U.K. His research interests include applied neural network research, computational geometry approaches towards neural networks, and knowledge extraction from neural networks.