PRACE NAUKOWE UNIWERSYTETU EKONOMICZNEGO WE WROCŁAWIU
Nr 7 (1207) 2008
TAKSONOMIA 15
Klasyfikacja i analiza danych – teoria i zastosowania

Esteban Alfaro Cortés, Matías Gámez Martínez, Noelia García Rubio
University of Castilla-La Mancha, Spain

A COMPARISON OF THREE ENSEMBLED CLASSIFIERS IN A FINANCIAL CASE

1. Introduction

In the last decades, several new classification methods based in the tree structure have been developed. We collect in this paper three of the most popular ensembles of trees, Adaboost, Bagging and Random Forest. AdaBoost [Freund and Schapire 1996] constructs its base classifiers in sequence, updating a distribution over the training examples to create each base classifier. Bagging [Breiman 1996] combines the individual classifiers built in bootstrap replicates of the training set. Finally, Random Forest [Breiman 2001] is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest.

We are interested in the corporate failure prediction task, an important management science problem [Alfaro, Gámez and García 2007]. Particularly, we focus on the search of a classifier as accurate as possible. The accuracy of the forecasting model is clearly of crucial importance in failure prediction because many economic agents are affected by the bankrupt 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.

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. We use a paired sample, using the activity and the size as control variables for this task. So, a healthy firm of the same sector and about the same size is jointed to each failed firm. This way the influence of these two features is avoided. The Legal structure, a qualitative variable, is also included as predictor in addition to the usual financial ratios.

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 Bagging and Random Forest methods.

Section 5 describes the data used in the analysis. The classification results are then presented and the classification models are compared on the basis of their prediction errors. Finally, following on from the empirical analysis, we present our conclusions.

2. AdaBoost

Boosting is a method that makes maximum using of a classifier by improving its accuracy. The classifier method is 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 using adaptive weights ($\omega_b(i)$). 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 [Banfield et al. 2004; Bauer and Kohavi 1999; Dietterich 2000; Freund and Schapire 1997].

Even though there are several versions of the boosting algorithms [Friedman, Hastie and Tibshirani 2000], the most widely used is the one by Freund and Schapire [1996] which is known as AdaBoost. For simplification purposes and without loss of generality, it can be assumed that there are only two classes. A training set is given $T_n = \{(X_1, Y_1), (X_2, Y_2), ...,(X_n, Y_n)\}$ where Y takes the values $\{-1, 1\}$. 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 and is applied to each training example. The error of this classifier is represented by ε_b and is calculated as

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

The complete AdaBoost algorithm is shown below:

AdaBoost Algorithm [Freund and Schapire 1996]

1. Start with $\omega_b(i) = 1/n$, $i = 1, 2, ..., n$.
2. Repeat for $b = 1, 2, ..., B$
   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 $C(x) = sign\left(\sum_{b=1}^{B} \alpha_b C_b(x)\right)$.

3. Bagging

Bagging is a method that combines bootstrapping and aggregating. If the bootstrap estimate of the data distribution parameters is more accurate and robust than the traditional one, then a similar method can be used to achieve, after combining them, a classifier with better properties.

On the basis of the training set, T_n, B bootstrap samples are obtained, where $b = 1, 2, ..., B$. These bootstrap samples are obtained by drawing with replacement the same number of elements than the original set (n in this case). In some of these samples, the presence of noisy observations may be eliminated or at least reduced, so the classifiers built in these sets will have a better behaviour than the classifier built in the original set. Therefore, bagging can be really useful to build a better classifier when there are noisy observations in the training set.

The ensemble classifier usually achieves better results than the single classifiers used to build the final classifier. This can be understood since combining the basic classifiers also combines the advantages of each one in the final classifier.

In particular the bagging method in the two class case where $Y \in \{-1, 1\}$ is applied in the following way:

Bagging Algorithm [Breiman 1996]

1. Repeat for $b = 1, 2, ..., B$
   a) Take a bootstrap replicate T^b of the training set T.
   b) Construct a single classifier $C_b(x)$ in T^b (with a decision boundary $C_b(x) = 0$).
2. Combine the basic classifiers $C_b(x)$, $b = 1, 2,..., B$ by the majority vote (the most often predicted class) to the final decision rule

$$\beta(x) = arg \max_{y \in \{-1,1\}} \sum_b \delta_{\mathrm{sgn}\left(C_b(x), y\right)} \quad where \quad \delta(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

4. Random Forest

Breiman [2001] defines a random forest as a classifier consisting of a collection of tree-structured classifiers $\{C(x, \Theta_i), i = 1, 2, ...\}$ where the $\{\Theta_i\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x.

Random Forest using random selection of features involves the joint use of two ensemble methods Bagging and Random Input Selection. The training sets are bootstrap samples of the same size as original drawn, with replacement, from the original data set. Then a new tree is built for each one of the training data set using random input selection. That is to say, in each node a small subset of features is randomly selected to split on. Then the tree is grown to maximum size and it is not pruned. The size F of the selected group of variables must be fixed previously.

Breiman tried two values of F. The first value was 1, so only one variable was used. The second took the first integer less than $\log_2 p + 1$, where p is the number of inputs. Later on, the same author advised to fix the F value as the square root of p, although according to him, the results were not sensitive to the number of features selected to split each node. From his experiments over twenty data sets commonly used in automatic learning, Breiman found surprisingly that using a single random input variable the results were only slightly worse or even better than selecting a group.

A random selection of features makes the procedure faster since the number of input variable for which the gain of information has to be calculated is reduced. So, constructing a random forest in this way will be quicker than bagging.

5. Data description

The companies in the sample were selected from the SABI database of Bureau Van Dijk which covers all the companies whose accounts are placed on the Spanish Mercantile Registry. 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 the previous year.

Healthy firms, on the other hand, were selected from active companies at the end of 2003 with full data for 2003 and 2002. Moreover any firm with constantly negative profits during the last three years would be rejected. For each distressed firm an active firm was selected having the same sector (NACE-93 3 digits) and similar size (lnTL). Within these requirements, 587 pairs of firms were selected (failed/healthy), obtaining 1174 observations for the total set.

Thirteen accounting-based ratios were included as predictors following the same criteria as in [Alfaro, Gámez and García 2007]. In addition to these accounting ratios, the legal structure was also used as a dichotomous variable. Only public corporations and limited corporations were considered because the number of other type of corporations was scarce. Therefore fourteen predictor variables were used for describing each company with information from the year prior to the moment of failure. These variables are listed in Table 1.

Table 1. 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
WC.TA	Working Capital/Total Assets	EBIT.CAP	Earnings before taxes/Capital
WC.S	Working Capital/Sales	S.CA	Sales /Current Assets
LE	Legal structure	S.TA	Sales /Total Assets

To detect the presence of outliers we use a multivariate approach. With this aim we compute the Mahalanobis distance of each example to the center of its class. We use only the numerical descriptors in this task. In our case, the Mahalanobis distance follows a Chi-squared distribution with 13 degrees of freedom. The critical value at a 99% of confidence is 27.688. Taking this into account, there are 43 failed firms and 48 healthy firms that we considered as outliers. In order to delete these outliers we should delete also their pairs to keep the paired structure of the sample. There are four pairs of examples where both are outliers so we have a final set of 500 pairs of healthy-failed firms, 1000 examples altogether.

6. Experimental results

In order to apply the methods above mentioned we have used the corresponding libraries in the R statistical program [R Development Core Team 2004]: rpart (single CART trees), adabag (AdaBoost and Bagging) and RandomForest (Random Forest). The R program is a freely available software program which can be downloaded from http://cran.r-project.org/.

In this paper we compare the accuracy of four classifiers, CART, Adaboost, Bagging and Random Forest. We should point out that Adaboost and Bagging can be used with other basic classifers, so here we use adaboosted trees or bagged trees, even we do not repeat it all the time to avoid being repetitive. Therefore, ahead in the paper when we mention Adaboost or Bagging we refer to adaboosted trees or bagged trees.

In order to ensure that comparison between the four tree classifiers does not happen by chance, we use five repetitions of 10-fold cross-validation [Opitz and Maclin 1999]. The entire set (1000 firms) is used for each 10-fold cross-validation experiment. This way we obtained the error rates of the four classifiers on each one of the 50 experiments.

Demsar [2006] is an important work dealing with the statistical comparisons of classifiers. According to this author, after checking the non normality of the error differences, we have used two non parametric methods to compare statistically the results of these four classifiers: the Wilcoxon signed-ranks test and counting of wins, losses and ties (sign test).

We check the normality for the distributions of the differences of errors yielded by each pair of classification methods using the tests shown in Table 2. As can be seen, in general, there is enough evidence to reject the normality hypothesis. This table also shows the average error and its standard deviation for each difference. Since the normality hypothesis is a fundamental requirement for the paired t-test, it has no sense applying it in this case.

Table 2. Normality tests for error differences

Error Difference	Kolmogorov-Smirnov		Shapiro			
	W	p-value	D	p-value	Average	Std. Dev.
CART – R.Forest	0.179789	0.078921	0.941521	0.015482	0.0106	0.014626
CART – Bagging	0.213612	0.020861	0.926945	0.004242	0.0018	0.013354
CART – Adaboost	0.120485	0.462355	0.962601	0.114249	0.0060	0.019692
R.Forest – Bagging	0.176238	0.089552	0.942213	0.016496	-0.0088	0.013192
R.Forest – Adaboost	0.185628	0.063757	0.927007	0.004265	-0.0046	0.017168
Bagging – Adaboost	0.139496	0.284912	0.967485	0.182532	0.0042	0.019069

The Wilcoxon signed-ranks test [Wilcoxon 1945] is a non-parametric alternative to the paired t-test when any of the require assumptions is not validated, which ranks the absolute value of the differences in performances of two classifiers for each data set and compares the ranks for the positive and the negative differences.

Table 3. Wilcoxon Signed-Ranks Test

Compared methods	Wilcoxon test		
	V	p-value	95% conf. interval
CART – R.Forest	795.5	1.585e-05	(0.009952,0.019927)
CART – Bagging	343	0.2610	(-7.138e-05,9.935e-03)
CART – Adaboost	576.5	0.0576	(-1.838e-05,1.507e-02)
R.Forest – Bagging	115.5	0.0002	(-0.015040,-0.009956)
R.Forest – Adaboost	237.5	0.0529	(-1.002e-02,2.827e-05)
Bagging – Adaboost	607	0.1886	(-3.070e-05,1.001e-02)

A popular way to compare the overall performance of classifiers is to count the number of data sets on which an algorithm is the overall winner and uses these counts with a form of binomial test, sign test [Sheskin 2000]. This test does not assume any commesurability of scores or differences nor does assume normal distribution and is thus applicable to any data set. Critical values for the two-tailed sign test at $\alpha = 0.05$ and $\alpha = 0.1$ are 32 and 31, respectively. A classifier is significantly better than another if it performs better at least 32 or 31 times. The order of the comparison is relevant and in case it turns, the value will be the complementary until the number of sets, 50 in this case. Therefore, a classifier is significantly worse than another if it performs better at the most 18 or 19 times.

Table 4. Sign Test

Compared methods	sign test
	wins+ties/2
CART – R.Forest	12
CART – Bagging	24
CART – Adaboost	18
R.Forest – Bagging	38
R.Forest – Adaboost	27
Bagging – Adaboost	21

Tables 3 and 4 show the superiority, for $\alpha = 0.05$, of Random Forest over CART and Bagging. In the case of the sign test also Adaboost outperforms CART at this level. The Wilcoxon test, at significance level of 10%, points out statistically significant differences between Random Forest and Adaboost, with the first ahead, and Adaboost against CART.

7. Conclusions

In this study, four classification methods have been compared. First, single trees have been applied with very satisfactory results (less than 10% of wrong classified cases) showing that classification trees are a powerful tool for corporate failure prediction. However, in terms of accuracy, the results may be improved by combining the response of a number of single trees.

As has been seen, combining methods improves in accuracy against single methods. Both Random Forest and Adaboost have shown statistically significant differences over single trees. Only the superiority of Bagging is not clear. Wilcoxon and ranked sign test show that, in this case, the best method is Random Forest, whose average differences are about 1.1 points with CART, 0.9 with Bagging and 0.5 with Adaboost.

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

References

Alfaro E., Gámez M. and García N. (2007), *A Boosting Approach for Corporate Failure Prediction*, „Applied Intelligence" 27, p. 29-37.

Banfield R.E., Hall L.O., Bowyer K.W., Bhadoria D., Kegelmeyer W.P. and Eschrich S. (2004), *A Comparison of Ensemble Creation Techniques*, [in:] F. Roli, J. Kittler and T. Windeatt (eds.), *Multiple Classifier Systems*, vol. 3077 of Lecture Notes in Computer Science, Springer, Cagliari, Italy, p. 223-232.

Bauer E., and Kohavi R. (1999), *An Empirical Comparison of Voting Classification Algorithm: Bagging, Boosting and Variants*, „Machine Learning" 36, p. 105-142.

Breiman L. (1996), *Bagging Predictors*, „Machine Learning" 24(2), p. 123-140.

Breiman L. (2001), *Random Forest*, „Machine Learning" 45, p. 5-32.

Breiman L., Friedman J.H., Olshen R., and Stone C.J. (1984), *Classification and Regression Trees*, Belmont, Wadsworth International Group.

Demsar J. (2006), *Statistical Comparison of Classifier over Multiple Data Sets*, „Journal of Machine Learning Research" 7, p. 1-30.

Dietterich T.G. (2000), *Ensemble Methods in Machine Learning*, [in:] J. Kittler and F. Roli (eds.), *Multiple Classifier Systems*, vol. 1857 of Lecture Notes in Computer Science. Springer, Cagliari, Italy, p. 1-15.

Freund Y. and Schapire R.E. (1996), *Experiments with a New Boosting Algorithm*, Proc. 13th International Conference on Machine Learning, Morgan Kaufmann, p. 148-146.

Freund Y. and Schapire R.E. (1997), A *Decision-theoretic Generalization of On-line Learning and an Application to Boosting*. „Journal of Computer and System Sciences" 55(1), p. 119-139.

Friedman J., Hastie T. and Tibshirani R. (2000), *Additive Logistic Regression: a Statistical View of Boosting*, „The Annals of Statistics" 38(2), p. 337-407.
Frydman H., Altman E. and Kao D. (1985), *Introducing Recursive Partitioning for Financial Classification: the Case of Financial Distress*, „Journal of Finance", p. 269-291.
Kuncheva L.I. (2004), *Combining Pattern Classifiers. Methods and Algorithms*, Wiley.
Opitz D. and Maclin R. (1999), *Popular Ensemble Methods: An Empirical Study*, „Journal of Artificial Intelligence Research" 11, p. 169-198.
R Development Core Team (2004), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Viena, http://www.R-project.org.
Sheskin D.J. (2000), *Handbook of Parametric and Nonparametric Statistical Procedures*, Chapman & Hall/CRC.
Wilcoxon F. (1945), *Individual Comparisons by Ranking Methods*, „Biometrics" 1, p. 80-83.

PORÓWNANIE TRZECH ZAGREGOWANYCH KLASYFIKATORÓW W ZAGADNIENIU FINANSOWYM

Streszczenie

Poczynając od lat 60., do predykcji upadłości stosuje się wiele technik klasyfikacyjnych. W stosunku do tradycyjnych modeli statystycznych drzewa klasyfikacyjne są narzędziem alternatywnym. Modele te są w stanie wychwycić nieliniowe zależności i wykazują dobre własności w przypadku obecności informacji jakościowych, co ma miejsce przy opisie sytuacji przedsiębiorstw, które poddawane są analizie w prognozowaniu bankructwa. Dlatego też drzewa klasyfikacyjne są szeroko wykorzystywane jako bazowe klasyfikatory przy budowie modeli zagregowanych. Celem tego badania jest porównanie zachowań trzech zagregowanych klasyfikatorów, tj. AdaBoost, Bagging and Random Forest w przypadku zastosowań w prognozowaniu upadłości.