*Microcomputers in Civil Engineering* **12** (1997) 267–276

# Using Machine Learning, Neural Networks, and Statistics to Predict Corporate Bankruptcy

P. P. M. Pompe*

*University of Twente, School of Management Science, 7500 AE Enschede, The Netherlands*

A. J. Feelders

*Data Distilleries, Kruislaan 419, 1098 VA Amsterdam, The Netherlands*

**Abstract:** *Recent literature strongly suggests that machine learning approaches to classification outperform "classical" statistical methods. We make a comparison between the performance of linear discriminant analysis, classification trees, and neural networks in predicting corporate bankruptcy. Linear discriminant analysis represents the "classical" statistical approach to classification, whereas classification trees and neural networks represent artificial intelligence approaches. A proper statistical design is used to be able to test whether observed differences in predictive performance are statistically significant. The data set consists of a collection of* 576 *annual reports from Belgian construction companies. We use stratified* 10*-fold cross-validation on the training set to choose "good" parameter values for the different learning methods. The test set is used to obtain an unbiased estimate of the true prediction error. Using rigorous statistical testing, we cannot conclude that in the case of the data set studied, one learning method clearly outperforms the other methods.*

## 1 INTRODUCTION

In the past decades, several researchers have developed statistical models for the prediction of corporate bankruptcy, e.g., Altman[1] and Bilderbeek.[2] A model for predicting corporate bankruptcy aims to describe the relation between bankruptcy and a number of explanatory financial ratios. These ratios can be calculated from the information contained in a company's annual report. The ultimate purpose is to obtain a method for timely prediction of bankruptcy, a so-called early warning system.

More recently, this subject has attracted the attention of researchers in the area of machine learning, e.g., Shaw and Gentry,[15] Fletcher and Goss,[7] and Tam and Kiang.[16] This research is usually directed at the comparison of machine learning methods, such as induction of classification trees and neural networks, with the "standard" statistical methods of linear discriminant analysis and logistic regression. Comparative studies of this kind in different domains (e.g., the StatLog Project[10]) show that some methods are better in some domains, whereas other methods are better in other domains.

In earlier research[6] we performed our first comparative analysis with bankruptcy data. The methods used were linear discriminant analysis, decision trees, and neural networks. We used a data set that contained 139 annual reports of Dutch industrial and trading companies. The experiments showed that the estimated prediction errors of both the decision tree and the neural network were below the estimated error of linear discriminant analysis. Thus it seems that we can gain by replacing the traditionally used linear discriminant analysis with a more flexible classification method to predict corporate bankruptcy. The data set used in these experiments was very small, however. To get more reliable results, we perform new experiments with a much larger and more recent set of Belgian annual reports.

This paper is organized as follows. In Section 2 we give some information about the data that are used. In Section 3 we give a short description of the three methods that will be compared in this study, namely, linear discriminant analy-

* To whom correspondence should be addressed.

**Table 1**
Balance sheet

| | | | | | |
|---|---|---|---|---|---|
| Fixed assets | ($p_1$) | ... | Capital contributed by shareholders | ($p_8$) | ... |
| Holdings | ($p_2$) | ... | Retained earnings | ($p_9$) | ... |
| Inventory | ($p_3$) | ... | Provisions | ($p_{10}$) | ... |
| Accounts receivable | ($p_4$) | ... | Long-term debt | ($p_{11}$) | ... |
| Other claims | ($p_5$) | ... | Short-term debt | ($p_{12}$) | ... |
| Cash | ($p_6$) | ... | Other debts | ($p_{13}$) | ... |
| | | | Accounts payable | ($p_{14}$) | ... |
| | | — | | | — |
| Total | ($p_7$) | ... | | ($p_7$) | ... |

sis, classification trees, and neural networks. In Section 4 we describe the design of our comparative experiment. Subsequently, the actual comparison, using a cross-validation approach, will be presented in Section 5. Finally, in Section 6 we draw some conclusions.

## 2 BELGIAN COMPANY DATA

We performed our experiments with a large set of Belgian annual reports. Since 1987, the National Bank of Belgium has been placing annual reports on CD-ROM. These CD-ROMs contain the reports of all companies that are legally obliged to publish their annual report. They contain information on about 175,000 companies. Using these CD-ROMs, we formed a collection that contains information on 576 construction companies. The information about each company consists of 10 financial ratios that were calculated using one annual report. In the collection, half the companies went "bankrupt." The ratios of bankrupt companies were calculated using the annual report published last but two before the bankruptcy. These companies went bankrupt in the period 1988 until 1994. Companies with less than five employees were not included in the collection.

In the collection, the other half of the companies belong to the class of "nonbankrupt." They were chosen randomly from the group of companies that published an annual report concerning the year 1990 and which did not go bankrupt in the period 1991–1994. The annual report concerning the the year 1990 was used to calculate the ratios. Again, companies with less than five employees were not included in the collection.

Many companies have parent-daughter relationships. It seems not wise to consider these companies as independent units because of the strong ties that exist with other companies. Experiments with a small part of the data set confirmed that including such companies negatively affects the predicion performance of the model. However, it was a small decrease of a few percent. Because the main goal of this study was to make a comparison between different learning techniques and a good comparison requires much data, we decided to include the companies with parent-daughter relationships.

The proportions of classes in the collection are not representative of the proportions in the population. In reality, the chance that a company will go bankrupt within a few years amounts to only a few percent. By taking into account the costs of misclassifications, the proportions used in the collection can be reasoned. The costs of misclassification of a company that belongs to the class "bankrupt" are much higher than the costs of misclassification of a company that belongs to the class "nonbankrupt." The exact relation between these costs depends on the specific application of a model. We assumed that the costs of misclassification of a company of class "bankrupt" divided by the costs of misclassification of a company of class "nonbankrupt" equal the proportion of class "nonbankrupt" divided by the proportion of class "bankrupt." This assumption justifies the proportion 0.5:0.5 that is used in the data collection. The advantage of not using the proportions as they occur in the population is that the collection does not need to be extremely large to guarantee enough instances of class "bankrupt."

The 10 ratios are common financial ratios. They include liquidity ratios, solvency ratios, profitability ratios, and activity ratios. An annual report consists of two parts, a balance sheet (see Table 1) and an income statement (see Table 2). Using one annual report, the 10 ratios can be calculated.

$$\begin{aligned} \text{r1} &= \frac{\text{working capital}}{\text{total assets}} \\ &= \frac{(p_3 + p_4 + p_5 + p_6) - (p_{12} + p_{13} + p_{14})}{p_7} \end{aligned}$$

$$\text{r2} = \frac{\text{accounts receivable}}{\text{added value}} = \frac{p_4}{p_{17}}$$

$$\text{r3} = \frac{\text{accounts payable}}{\text{added value}} = \frac{p_{14}}{p_{17}}$$

**Table 2**
Income statement

| | | | |
|---|---|---|---|
| Sales | ($p_{15}$) | ... | |
| Cost of goods and services bought | ($p_{16}$) | ... | − |
| | | — | |
| Added value | ($p_{17}$) | ... | |
| Salaries | ($p_{18}$) | ... | − |
| Other operating expenses | ($p_{19}$) | ... | − |
| | | — | |
| Gross earnings | ($p_{20}$) | ... | |
| Depreciation | ($p_{21}$) | ... | − |
| Interest | ($p_{22}$) | ... | − |
| Other income/costs | ($p_{23}$) | ... | ± |
| | | — | |
| Earnings before taxes | ($p_{24}$) | ... | |
| Taxes | ($p_{25}$) | ... | − |
| | | — | |
| Net profit | ($p_{26}$) | ... | |

$$\text{r4} = \frac{\text{equity}}{\text{total assets}} = \frac{p_8 + p_9}{p_7}$$

$$\text{r5} = \frac{\text{retained earnings}}{\text{total assets}} = \frac{p_9}{p_7}$$

$$\text{r6} = \frac{\text{net profit}}{\text{added value}} = \frac{p_{26}}{p_{17}}$$

$$\text{r7} = \frac{\text{earnings before taxes and interest}}{\text{total assets}} = \frac{p_{24} + p_{22}}{p_7}$$

$$\text{r8} = \frac{\text{net profit}}{\text{equity}} = \frac{p_{26}}{p_8 + p_9}$$

$$\text{r9} = \frac{\text{added value}}{\text{total assets}} = \frac{p_{17}}{p_7}$$

$$\text{r10} = \frac{\text{salaries}}{\text{added value}} = \frac{p_{18}}{p_{17}}$$

Because a large number of companies are not obliged to report the value of sales, ratios that need this value to be calculated are not used.

Regarding the selection of variables, it is wise to choose the 10 ratios instead of the 26 elements ($p_1, p_2, \ldots, p_{26}$) of the annual report because the ratios contain domain knowledge of financial experts. The 10 ratios were appointed by financial experts as being important variables in the case of the prediction of bankruptcy.

In this domain we do not expect a very low prediction error for any of the models generated by the learning methods. A specific bankruptcy has many causes. Not every cause influences the information in the annual report. Furthermore, a portion of the causes occurs in the years after the moment of prediction.

## 3 CLASSIFICATION METHODS

In this section three methods for learning classification rules are described very briefly. In Sec. 5 these methods are evaluated on the Belgian data.

### 3.1 Linear discriminant analysis

Linear discriminant analysis is probably the most popular "classical" statistical method of classification. It is most easily viewed as a special case of the well-known Bayes criterion. This criterion states that assuming one wants to minimize the number of misclassifications, to assign an object to group $C_j$ if

$$P(C_j)\, P(\mathbf{x} \mid C_j) > P(C_i)\, P(\mathbf{x} \mid C_i) \qquad \forall i \neq j$$

where $P(C_j)$ is the relative frequency of $C_j$ in the population, and $P(\mathbf{x} \mid C_j)$ is the conditional probability of observing feature vector $\mathbf{x}$, given that an object belongs to group $C_j$.

Although this rule is optimal, its straightforward application requires the estimation of many conditional probabilities. In order to reduce this problem, statisticians have introduced additional assumptions. If it is assumed that $\mathbf{x}$ follows a multivariate normal distribution in all groups and all groups have identical covariance matrices, then the preceding rule can be replaced by assigning to group $C_j$ if

$$f_j(\mathbf{x}) + \ln(P(C_j)) > f_i(\mathbf{x}) + \ln(P(C_i)) \qquad \forall i \neq j$$

where

$$f_i(\mathbf{x}) = \mu_i' \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mu_i' \Sigma^{-1} \mu_i$$

where $\Sigma$ denotes the covariance matrix common to the multivariate normal distribution of $\mathbf{x}$ in all groups, and $\mu_i$ denotes the population mean vector in group $C_i$.

If it is additionally assumed that objects are to be classified into one of two groups, the optimal rule is to classifiy into group 1 if

$$z(\mathbf{x}) > \ln(P(C_2)) - \ln(P(C_1))$$

and to group 2 otherwise, where $z(\mathbf{x}) = f_1(\mathbf{x}) - f_2(\mathbf{x})$. This is the form in which most discriminant functions are presented in the literature.

In practice, the population parameters $\mu_i$ and $\Sigma$ are unknown and have to be estimated from the sample. The discriminant function is estimated using the sample estimates $\overline{\mathbf{x}}_\mathbf{i}$ and the pooled covariance matrix $S_{\text{pooled}}$. This is often called the *plug-in estimate* of the discriminant function.

In linear discriminant analysis, a stepwise method to reduce the number of variables in the discriminant functions is often used. In a stepwise method, the first variable included in the analysis has the largest acceptable value for the selection criterion. After the first variable is entered, the value of the criterion is reevaluated for all variables not in the model, and the variable with the largest acceptable criterion value is entered next. At this point, the variable entered first is reevaluated to determine whether it meets the removal criterion. If it does, it is removed from the model. The next step is to examine the variables not in the model for entry, followed by examination of the variables in the model for removal. Variables are removed until none remain that meet the removal criterion. Variable selection terminates when no more variables meet the entry or removal criterion. Whether a variable meets the entry or removal criterion depends on the significance of the change in the selection criterion when the variable is entered or removed from the model. The significance of the change is based on an $F$ statistic. The selection criterion we used is the Mahalanobis distance, a generalized measure of the distance between two groups. The distance between groups $i$ and $j$ is defined as

$$(\mu_i - \mu_j)' \Sigma^{-1} (\mu_i - \mu_j)$$

### 3.2 Classification trees

We already noted that linear discriminant analysis is based on a number of assumptions that may not always be realistic, e.g., when data are not continuous interval scaled. A number of alternative methods have been developed (stimulated by increased computer power) that make no such a priori assumptions. Examples of such *nonparametric* methods are classification trees, neural networks, and $k$-nearest neighbor.

The basic idea of classification trees is as follows. A tree is fitted to the training sample by *recursive partitioning*, which means that the training sample is successively split into increasingly homogeneous subsets until the leaf nodes contain only cases from a single class or some other reasonable stopping criterion applies. Well-known examples of classification tree algorithms are CART,[3] ID3,[11] and its successor C4.5.[13] Classification trees approximate the optimal Bayes classifier by making a partition of the sample space and estimating the posterior probabilities (and hence the Bayes rule) within each cell of the partition by the relative frequencies of the classes of the training cases that fall within that cell.[14]

A crucial decision in building a classification tree is selecting the variable on which to make the next split. Since one strives toward homogeneous subsets, most algorithms employ some measure to indicate the "impurity" of a set of cases, i.e., the extent to which a node contains training cases from multiple classes. Such a measure should take its largest value when all classes are equally represented and its smallest value when the node contains members of a single class. For example, CART employs the so-called Gini-index

$$i(t) = \sum_{j \neq k} P(j \mid t) P(k \mid t)$$

where $i(t)$ denotes the impurity at node $t$, and $P(j \mid t)$ denotes the proportion of cases from class $j$ at node $t$. On the other hand, ID3 uses the *entropy* of a node for the same purpose:

$$i(t) = -\sum_{j=1}^{n} P(j \mid t) \times \log_2(P(j \mid t))$$

where $n$ is the number of classes at node $t$.

In this study we use the function *tree* of the data analysis program Splus for building a classification tree. This function employs the so-called deviance of a node[17] as the measure of impurity. The *deviance* is defined as follows.

$$i(t) = -2 \sum_{j=1}^{n} N(j \mid t) \log(P(j \mid t))$$

Here $N(j \mid t)$ denotes the number of cases of class $j$ at node $t$.

In CART and ID3, the "optimal" split is the one for which

$$\sum_{k=1}^{m} P(t_k) i(t_k)$$

is minimal, where $m$ is the number of subsets resulting from the split, and where $P(t_k)$ is the proportion of cases in subset $t_k$.

The Splus *tree* function chooses the optimal split by minimizing

$$\sum_{k=1}^{m} i(t_k)$$

The Splus *tree* function and ID3 always result in the same optimal split.

Another critical issue in learning classification trees and other flexible classification methods is the prevention of overfitting on the training sample. To this end, one usually adopts a pruning strategy, whereby the tree is first grown to its full size and then leaf nodes are merged back or "pruned" to produce a smaller tree. Examples of pruning strategies are CART's cost-complexity pruning[3] and reduced-error pruning.[12]

We use the function *prune.tree* of Splus for pruning a classification tree. This function uses cost-complexity pruning to prevent overfitting. The full-size tree is pruned in such a way the value of $R + \alpha$ (size) is minimized,[17] where the size of a tree is equal to the number of leaves, and $R$ is the sum of deviances of the leaves. The value of $\alpha$ is chosen by the user.

### 3.3 Neural networks

Recently, neural networks have become a very popular means for learning classification functions.[7,16] It is beyond the scope

of this paper to discuss the many types of networks found in the literature. Instead, we restrict our attention to the type of network actually used in this study. This type is called a *feedforward network*[8] with a single hidden layer.

In a feedforward network the input units (independent variables) are connected to the "hidden" units in the second layer. With the connection between input node $i$ and hidden node $h$, a weight $w_{ih}$ is associated. The hidden units sum their inputs ($x_i \times w_{ih}$), add a constant (called *bias*), and take a fixed function $f_h$ of the result. The output units are of the same form, but with output function $f_o$. In the following we assume there is only one output unit. Consequently, output $y$ is defined as

$$y = f_o\left[\alpha_o + \sum_h w_{ho}\, f_h\left(\alpha_h + \sum_i w_{ih}\, x_i\right)\right]$$

For classification problems, the activation functions $f_h$ and $f_o$ are usually taken to be the logistic function

$$\ell(z) = \frac{e^z}{1+e^z}$$

The weights $w_{ij}$ have to be chosen to minimize some fitting criterion. The measure used in most classification studies is cross-entropy, which amounts to minimizing

$$E = \sum_p \left[t^p \log\frac{t^p}{y^p} + (1-t^p)\log\frac{1-t^p}{1-y^p}\right]$$

where $t^p$ is the target and $y^p$ the output for the $p$th example pattern.

We already mentioned that overfitting is a general problem for flexible discrimination methods. A heuristic to avoid overfitting in neural networks is *weight decay*. Weight decay amounts to minimizing

$$E + \lambda \sum_{ij} w_{ij}^2$$

where $\lambda$ denotes the weight decay parameter. Intuitively, this can be viewed as a way to penalize large weights, thus ensuring smoother fits.

## 4 DESIGN OF THE COMPARATIVE EXPERIMENT

The main goal of this study is to make a comparison between different learning techniques for predicting corporate bankruptcy. To this end, we have chosen the following design.

The collection of construction companies is randomly split into two subsets of 288 instances each, a training set and a test set. Both subsets are stratified; i.e., the proportions of classes in a subset equal the proportions in the original data set. To get more reliable test results, half the data instead of the common one third is used for testing, assuming that the other half is a reasonable amount of training data for each of the learning methods. We use stratified 10-fold cross-validation on the training set to choose "good" parameter values for the different learning methods. In 10-fold cross-validation, the training set $D$ is randomly split into 10 subsets $D_1, D_2, \ldots, D_{10}$ of (approximately) equal size. The model is trained and tested 10 times; each time $t = 1, 2, \ldots, 10$, it is trained on $D/D_t$ and tested on $D_t$. The cross-validation error is the overall number of incorrect classifications divided by the number of instances in the training set.

The same 10 subsets in cross-validation are used each time a value or, in case of more than one parameter, a combination of values is tried. This is done in order to rule out variations due to the random selection of subsets. The parameter value or combination of parameter values that yields the lowest cross-validation error is taken to be optimal. However, this cross-validation error estimate is optimistically biased. Therefore, a new model is trained using the whole training set with the optimal parameter setting(s) found in the cross-validation experiment. Subsequently, the test set is used to obtain an unbiased estimate of the true prediction error of this model. The true prediction error is the number of incorrect classifications in the population divided by the number of instances in the population.

To get more reliable results, we performed the whole experiment 10 times.

## 5 COMPARISON OF RESULTS

### 5.1 Linear discriminant analysis

For the linear discriminant analysis, we used the *lda* and *predict.lda* functions of Splus.[17] The stepwise selection of variables was performed by a Splus function we programmed ourselves. In the analysis, the parameters used were the prior probabilities of class membership and the number of variables in the discriminant function. Regarding the prior probabilities (prior probability "nonbankrupt": prior probability "bankrupt"), we experimented with 11 values: (0.40:0.60), (0.42:0.58), (0.44:0.56), . . . , (0.60:0.40). Regarding the number of variables, 2 values were tried: all 10 variables and the reduced group after stepwise selection on $D/D_t$. Thus in each of the 10 experiments $11 \times 2 = 22$ different parameter settings were tried. In Table 3 the lowest cross-validation error, the parameter settings that led to that result, and the prediction error for the test set are shown. If two parameter settings gave the same lowest cross-validation error, the settings with stepwise selection, i.e., the settings that lead to the most simple function, were chosen. As an example, a discriminant function is shown in Fig. 1. It is the function trained using the whole training set with the optimal parameter settings in experiment 1.

**Table 3**
For each experiment and each learning method: the lowest cross-validation error, the parameter setting(s) that led to that result, and the test set error

| | *Linear discriminant* | *Classification tree* | *Neural network* |
|---|---|---|---|
| Experiment 1 | | | |
| Parameter settings | (0.52:0.48)/all | 9 | 2/0.05 |
| Cross-validation | 0.285 | 0.295 | 0.264 |
| Test set | 0.340 | 0.337 | **0.319** |
| Experiment 2 | | | |
| Parameter settings | (0.56:0.44)/step | 15 | 1/0.25 |
| Cross-validation | 0.323 | 0.385 | 0.316 |
| Test set | 0.302 | 0.399 | **0.274** |
| Experiment 3 | | | |
| Parameter settings | (0.52:0.48)/all | 11 | 4/0.25 |
| Cross-validation | 0.285 | 0.333 | 0.264 |
| Test set | 0.330 | 0.337 | **0.312** |
| Experiment 4 | | | |
| Parameter settings | (0.50:0.50)/all | 13 | 2/0.05 |
| Cross-validation | 0.302 | 0.326 | 0.267 |
| Test set | **0.306** | 0.323 | 0.323 |
| Experiment 5 | | | |
| Parameter settings | (0.52:0.48)/all | 6 | 5/0.05 |
| Cross-validation | 0.292 | 0.312 | 0.292 |
| Test set | **0.333** | 0.413 | 0.337 |
| Experiment 6 | | | |
| Parameter settings | (0.40:0.60)/step | 9 | 5/0.15 |
| Cross-validation | 0.323 | 0.351 | 0.312 |
| Test set | 0.344 | 0.340 | **0.285** |
| Experiment 7 | | | |
| Parameter settings | (0.48:0.52)/all | 6 | 2/0.15 |
| Cross-validation | 0.306 | 0.351 | 0.295 |
| Test set | 0.323 | 0.378 | **0.285** |
| Experiment 8 | | | |
| Parameter settings | (0.46:0.54)/all | 11 | 4/0.15 |
| Cross-validation | 0.347 | 0.319 | 0.309 |
| Test set | 0.288 | 0.312 | **0.285** |
| Experiment 9 | | | |
| Parameter settings | (0.56:0.44)/step | 13 | 5/0.1 |
| Cross-validation | 0.309 | 0.281 | 0.271 |
| Test set | 0.372 | 0.347 | **0.323** |
| Experiment 10 | | | |
| Parameter settings | (0.58:0.42)/all | 10 | 5/0.05 |
| Cross-validation | 0.319 | 0.354 | 0.309 |
| Test set | 0.340 | 0.299 | **0.292** |

### 5.2 Classification trees

In our study, the learning and pruning of classification trees were performed using the *tree* function and the *prune.tree* function of the Splus. In each experiment we tried 26 different values $(0, 1, 2, \ldots, 25)$ for the parameter $\alpha$ that determines the amount of pruning. In Table 3 the lowest cross-validation error, the value of $\alpha$ that led to that result, and the prediction

```
If  ( ratio1 x 2.08  +  ratio2 x -0.00082  +  ratio3 x -0.44  +  ratio4 x 1.46  +
      ratio5  x  -0.21  +  ratio6 x 2.24  + ratio7 x 3.12  +  ratio8  x  0.0019  +
      ratio9  x  -1.27  +  ratio10  x  1.17  +  -0.44 )  >  0
then 'non-bankrupt',  else  'bankrupt'
```

**Fig. 1.** Linear discriminant function trained using the whole training set in experiment 1.

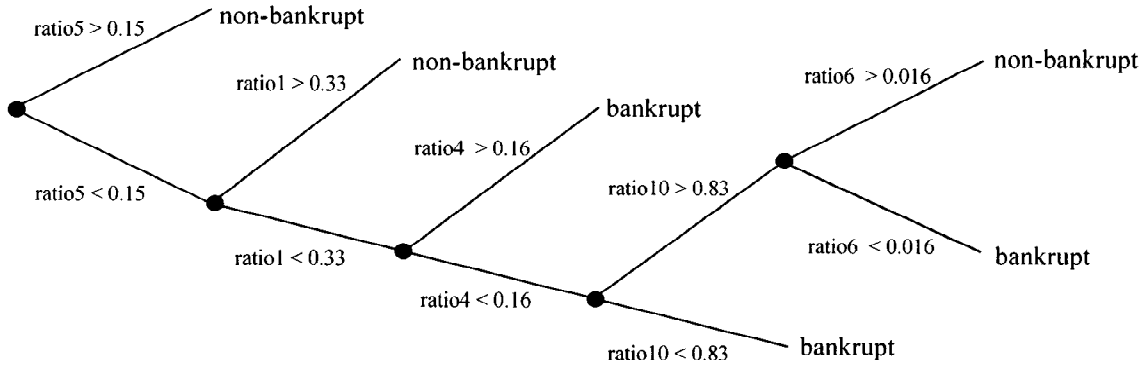



**Fig. 2.** Classification tree trained using the whole training set in experiment 1.

error for the test set for each experiment are shown. If two values gave the same lowest cross-validation error, we chose the highest value, i.e., the value that leads to the most simple tree with the highest degree of pruning. Figure 2 shows the classification tree in experiment 1 that is trained using the whole training set with the optimal degree of pruning.

### 5.3 Neural networks

For the neural network analysis, we used the *nnet* and *predict.nnet* functions of Splus.[17] This is a single-hidden-layer, feedforward network with the possibility to use weight decay to prevent overfitting. We used cross-entropy as the error function and included skip-layer connections, which means that the network encompasses a linear model. Before performing the experiments, we rescaled the 10 ratios. After rescaling a ratio, 90% of its values in the data set were within the interval [0, 1].

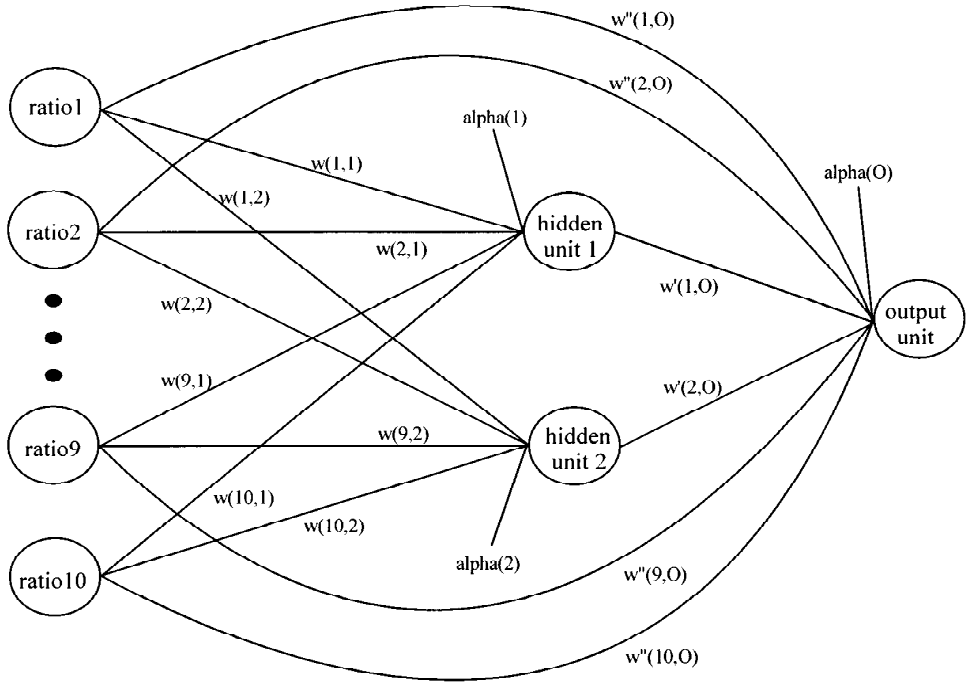
For neural networks, the parameters used were the number of hidden units (0, 1, 2, . . . , 6), the amount of weight decay (0, 0.05, 0.1, 0.15, 0.2, 0.25), and the set of starting weights (set1, set2, set3, set4, set5). In each experiment and for each combination (number of hidden units, amount of weight decay), 5 sets of random starting weights were generated. These 5 sets were the 5 values of the third argument. Thus in each experiment we tried $7 \times 6 \times 5 = 210$ different parameter settings. Again, the results can be found in Table 3. Of the parameter settings that led to the lowest cross-validation error, only the number of hidden units and the amount of weight decay are shown. If two parameter settings gave the same lowest cross-validation error, the settings that lead to the most simple network were chosen. These are the settings with the lowest number of hidden units and (if for two settings this number is equal) the settings with the highest amount of weight decay. Figure 3 shows the neural network in experiment 1 that is trained using the whole training set with the optimal parameter settings.

From a domain-expert perspective, the classification tree seems to be most interesting, because the model is comprehensible. The neural network is not comprehensible at all.

### 5.4 Comparison

In this subsection we make a comparison between the best linear discriminant function, classification tree, and neural network for each of the 10 experiments. Henceforth, these functions will be denoted by $f_1$, $f_2$, and $f_3$, respectively. The corresponding true prediction error is denoted by $\theta_i$. For the underlying ideas of the methods of comparison used, we refer the reader to Feelders and Verkooijen.[5]

For each experiment we perform all $k^* = 3$ pairwise comparisons. In each comparison the null hypothesis is $\theta_i = \theta_{i'}$ and the alternative hypothesis is $\theta_i \neq \theta_{i'}$. We use a multiple-comparison procedure, which takes into account the dependence between the $k^*$ comparisons and allows for controlling the familywise error rate ($\alpha$), i.e., the probability of making one or more type I errors within the group of $k^*$ comparisons (a type I error means incorrectly rejecting the null hypothesis). Table 4 presents the general layout of a study that compares $k$ classification functions. In this matrix, $Y_{ji}$ has the value zero if $f_i$ classifies observation $j$ correctly and one otherwise. In our study, observation $j$ is instance $j$ in the test set.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| w(i,1) | 0.92 | -0.09 | -2.05 | 0.10 | 0.15 | 1.78 | 0.04 | 3.03 | 0.23 | 1.19 |
| w(i,2) | -2.75 | -1.25 | 0.80 | 2.39 | 1.12 | -0.51 | 2.67 | -0.46 | 2.10 | -0.07 |

| | 1 | 2 |
|---|---|---|
| w'(i,O) | 3.40 | -6.33 |

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| w"(i,O) | 0.07 | -1.37 | -0.09 | 2.40 | 0.87 | 1.68 | 0.78 | -0.19 | -1.72 | 0.87 |

| | 1 | 2 | O |
|---|---|---|---|
| alpha(i) | -0.42 | -1.35 | -0.58 |

**Fig. 3.** Neural network trained using the whole training set in experiment 1.

**Table 4**
Data table for comparison of classification functions

| *Observations* | *Functions* $f_1$ | $f_2$ | ... | $f_i$ | ... | $f_k$ | Total |
|---|---|---|---|---|---|---|---|
| 1 | $Y_{11}$ | $Y_{12}$ | ... | $Y_{1i}$ | ... | $Y_{1k}$ | $Y_{1.}$ |
| 2 | $Y_{21}$ | $Y_{22}$ | ... | $Y_{2i}$ | ... | $Y_{2k}$ | $Y_{2.}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $j$ | $Y_{j1}$ | $Y_{j2}$ | ... | $Y_{ji}$ | ... | $Y_{jk}$ | $Y_{j.}$ |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $n$ | $Y_{n1}$ | $Y_{n2}$ | ... | $Y_{ni}$ | ... | $Y_{nk}$ | $Y_{n.}$ |
| Total | $Y_{.1}$ | $Y_{.2}$ | ... | $Y_{.i}$ | ... | $Y_{.k}$ | |
| Means | $\overline{Y}_{.1}$ | $\overline{Y}_{.2}$ | ... | $\overline{Y}_{.i}$ | ... | $\overline{Y}_{.k}$ | |

**Table 5**
Confidence intervals for all pairwise differences in all experiments

| *Experiment* | $\theta_{ldiscr} - \theta_{tree}$ | $\alpha$ | $\theta_{ldiscr} - \theta_{nnet}$ | $\alpha$ | $\theta_{tree} - \theta_{nnet}$ | $\alpha$ |
|---|---|---|---|---|---|---|
| 1 | [–0.045, 0.052] | 0.20 | [–0.027, 0.069] | 0.20 | [–0.031, 0.065] | 0.20 |
| 2 | **[–0.18, –0.016]** | **0.01** | [–0.023, 0.079] | 0.20 | **[0.044, 0.21]** | **0.01** |
| 3 | [–0.046, 0.032] | 0.20 | [–0.022, 0.056] | 0.20 | [–0.015, 0.063] | 0.20 |
| 4 | [–0.061, 0.026] | 0.20 | [–0.061, 0.026] | 0.20 | [–0.044, 0.044] | 0.20 |
| 5 | **[–0.16, –0.0038]** | **0.05** | [–0.062, 0.055] | 0.20 | **[0.00033, 0.15]** | **0.05** |
| 6 | [–0.052, 0.058] | 0.20 | **[0.00075, 0.12]** | **0.15** | **[0.00055, 0.11]** | **0.20** |
| 7 | **[–0.11, –0.0021]** | **0.20** | [–0.015, 0.092] | 0.20 | **[0.0087, 0.18]** | **0.01** |
| 8 | [–0.062, 0.014] | 0.20 | [–0.035, 0.042] | 0.20 | [–0.010, 0.066] | 0.20 |
| 9 | [–0.012, 0.083] | 0.20 | [–0.0028, 0.092] | 0.20 | [–0.014, 0.081] | 0.20 |
| 10 | [–0.0097, 0.093] | 0.20 | [–0.0028, 0.10] | 0.20 | [–0.044, 0.058] | 0.20 |

The pooled variance of any pairwise difference $\overline{Y}_{.i} - \overline{Y}_{.i'}$ for this design can be written as[9]

$$\hat{\sigma}^2_{\text{diff}} = \frac{2(k\Sigma_{j=1}^{n} Y_{j.} - \Sigma_{j=1}^{n} Y_{j.}^2)}{n^2 k(k-1)}$$

We can now construct $100(1-\alpha)\%$ simultaneous confidence intervals for all pairwise differences $\theta_i - \theta_{i'}$ as follows:

$$\theta_i - \theta_{i'} \in \left[\overline{Y}_{.i} - \overline{Y}_{.i'} \pm Z^{\nu}_{k^*:1-\alpha/2}\hat{\sigma}_{\text{diff}}\right] \qquad (1 \le i < i' \le k)$$

where $\nu = n-1$ denotes degrees of freedom. The distribution of $Z$ is based on the student $t$ distribution, adjusted for the number of comparisons $k^*$ involved.[4] Tables for this statistic can be found in refs. 4 and 9. As one would expect, the value of $Z^{\nu}_{k^*:1-\alpha/2}$ increases with the number of comparisons $k^*$, leading to wider confidence intervals.

If the interval of $\theta_i - \theta_{i'}$ contains zero, then there is no significant evidence that classification functions $f_i$ and $f_{i'}$ differ in their true prediction error. We have calculated confidence intervals for each pairwise difference in each experiment for the following values of $\alpha$: 0.01, 0.05, 0.10, 0.15, and 0.20. Table 5 provides for each pairwise difference the lowest value of $\alpha$ (and the corresponding interval) for which the interval does not contain zero. If none of the values of $\alpha$ leads to an interval that does not contain zero, Table 5 shows the interval at $\alpha = 0.20$.

From this table we infer that the following significant differences have been found:

- Between the linear discriminant and the classification tree in experiment 2 (at $\alpha = 0.01$), in experiment 5 (at $\alpha = 0.05$), and in experiment 7 (at $\alpha = 0.20$)
- Between the linear discriminant and the neural network in experiment 6 (at $\alpha = 0.15$)
- Between the classification tree and the neural network in experiment 2 (at $\alpha = 0.01$), in experiment 5 (at $\alpha = 0.05$), in experiment 6 (at $\alpha = 0.20$), and in experiment 7 (at $\alpha = 0.01$)

## 6 CONCLUSIONS

Looking at Table 3, the first impression is that neural networks perform better than linear discriminant analysis and classification trees. For the 10 experiments, neural networks have the lowest prediction error for the test set 8 times.

However, on rigorous statistical testing using a multiple-comparisons procedure, only a few significant differences have been found. We cannot conclude that one learning technique clearly outperformed the other techniques. Naturally, this conclusion is specific to the data set studied. It seems that the data used satisfy the assumptions of linear discriminant analysis to an extent that the nonparametric methods classification trees and neural networks are not able to improve on significantly.

For the best models generated in this study, the value of (*the true prediction error for class "bankrupt" + the true prediction error for class "nonbankrupt"*):2 is around 0.3. It would be interesting to investigate if domain experts can predict at the same performance level as these models.

## REFERENCES


1. Altman, E. I., Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, *The Journal of Finance*, **23** (4) (1968), 589–609.
2. Bilderbeek, J., *Financiële ratio-analyse*, Stenfert Kroese, Leiden, 1983.
3. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. T., *Classification and Regression Trees*, Wadsworth, Belmont, CA, 1984.
4. Dunn, O. J., Multiple comparisons among means, *Journal of the American Statistical Association*, **56** (1961), 52–64.
5. Feelders, A. & Verkooijen, W., Which method learns most from the data? in *Proceedings of the Fifth International Workshop on AI and Statistics*, Miami, D. Fisher & H. Lenz, eds.,1995, pp. 219–25.

6. Feelders, A., Pompe, P. & Vet, P. E. v.d., The twente-test: A comparison of machine learning and statistics for predicting corporate bankruptcy, in *Proceedings of BENELEARN*-94, Rotterdam, J. Bioch & S. Nienhuys-Cheng, eds., Report EUR-CS-94-05, 1994, pp. 180–91.
7. Fletcher, D. & Goss, E., Forecasting with neural networks: An application using bankruptcy data, *Information and Management*, **24** (1993), 159–67.
8. Hertz, J., Krogh, A. & Palmer, R. G., *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA, 1991.
9. Marascuilo, L. & McSweeney, M., *Nonparametric and Distribution-Free Methods for the Social Sciences*, Brooks/Cole, Monterey, CA, 1977, p. 180.
10. Michie, D., Spiegelhalter, D. J. & Taylor, C. C. (eds.), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, New York, 1994.
11. Quinlan, J., Learning efficient classification procedures, in *Machine Learning*: *An Artificial Intelligence Approach*, R. Michalski, J. Carbonell & T. Mitchell, eds., Tioga Press, Palo Alto, CA, 1983.
12. Quinlan, J., Simplifying decision trees, *International Journal of Man-Machine Studies*, **27** (1987), 221–34.
13. Quinlan, J., *C*4.5 *Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA, 1993.
14. Ripley, B. D., Flexible non-linear approaches to classification, in *From Statistics to Neural Networks*. *Theory and Pattern Recognition Applications*, V. Cherkassky, J. Friedman & H. Wechsler, eds., Springer-Verlag, Berlin, 1994.
15. Shaw, M. J. & Gentry, J. A., Inductive learning for risk classification, *IEEE Expert*, (1990), 47–53.
16. Tam, K. Y. & Kiang, M. Y., Managerial applications of neural networks: The case of bank failure predictions, *Management Science*, **38** (7) (1992), 926–47.
17. Venables, W. & Ripley, B., *Modern Applied Statistics with S-Plus*, Springer-Verlag, New York, 1994.