A Genetic Programming Approach for Bankruptcy Prediction Using a Highly Unbalanced Database

Eva Alfaro-Cid, Ken Sharman, and Anna Esparcia-Alcázar

Instituto Tecnológico de Informática, Universidad Politécnica de Valencia,
Camino de Vera s/n, 46022 Valencia, Spain

Abstract. In this paper we present the application of a genetic programming algorithm to the problem of bankruptcy prediction. To carry out the research we have used a database of Spanish companies. The database has two important drawbacks: the number of bankrupt companies is very small when compared with the number of healthy ones (*unbalanced data*) and a considerable number of companies have missing data. For comparison purposes we have solved the same problem using a support vector machine. Genetic programming has achieved very satisfactory results, improving those obtained with the support vector machine.

1 Introduction

Bankruptcy prediction is a very important economic issue. It is of great significance for stakeholders as well as creditors, banks, investors, to be able to predict accurately the financial distress of a company. Given its relevance in real life, it has been a major topic in the economic literature. Many researchers have worked on this topic during the last decades; however, there is no generally accepted prediction model.

According to [7], a survey reviewing journal papers on the field in the period 1932-1994, the most popular methods for building quantitative models for bankruptcy prediction have been discriminant analysis [1] and logit analysis [16]. Since the 90s there has been an increasing interest on the application of methods originating from the field of artificial intelligence, mainly neural networks [14].

However, other methods from the artificial intelligence field, such as evolutionary computation have been scarcely used for the bankruptcy prediction problem. After an extensive (but not exhaustive) review of the literature, only a few papers could be found that applied evolutionary methods to the bankruptcy prediction problem.

Some authors have used genetic algorithms (GAs), either on its own [11], [18], [20] or in a hybrid method with a neural network [4] for the insolvency prediction problem. However, most of the approaches from the evolutionary computation field use genetic programming (GP). The ability of GP to build functions make this algorithm more appropriate to the problem at hand than GA. In the literature we can find a couple of hybrid approaches that combine GP with another

M. Giacobini et al. (Eds.): EvoWorkshops 2007, LNCS 4448, pp. 169–178, 2007.

method, such as rough sets [15] and neural networks [19]. Some authors have used GP on its own. In [21], the authors have used linear GP and have compared its performance to support vector machines and neural networks. In [13], the authors have used GP to predict bankruptcy on a database of Norwegian companies and in [17] the GP has been used for the prediction of insolvency in non-life insurance companies, a particular case. Finally, grammatical evolution, a form of grammar-based genetic programming, has been used in [5] to solve several financial problems.

One important advantage of the GP approach to bankruptcy prediction is that it yields the rules relating the measured data to the likelihood of becoming bankrupt. Thus a financial analyst can see what variables and functions thereof are important in predicting bankruptcy.

Our approach differs from previous GP applications in the characteristics of the database we are using. Our database comprises data from Spanish companies from different industrial sectors. The database has two drawbacks: firstly, it is highly unbalanced (only 5-6% of the companies go bankrupt) and, secondly, some data are missing. Although this complicates the classification, it is an accurate reflection of the real world, where few companies go bankrupt in proportion and it is difficult to obtain all the relevant data from companies.

For comparison we have also analyzed the data using a support vector machine (SVM) classifier, and our results demonstrate that our proposed GP technique gives improved performance over the SVM.

2 Financial Data

The work presented in the paper uses a database supplied by the Department of Finance and Accounting of the Universidad de Granada, Spain.

The database consist of a 2859×31 matrix comprising data from 484 Spanish companies from the year 1998 to the year 2003[1]. Each row of the matrix holds the data referent to a company during one year. The database includes not only financial data such as solvency, profit margin or income yield capacity, but also general information such as company size, age or number of partners. These variables are the inputs to the classifier. The desired output of the classifier is the variable that states if the company was bankrupt in 2003 or not.

In this work we have used the data from years 1999 and 2000 to predict bankruptcy in the year 2003, that is 4 and 3 years in advance, respectively.

All variables can take values from different numerical ranges. Some of them take real values, others take real positive values, others take integer values and, finally there are four boolean variables that indicate if the company has been audited, if there was any delay in presenting the accountancy, if the company is linked to a group or if the company has been suffering losses. Therefore, as the numerical range the variables can take varies a lot, all the data have been

[1] The number of rows in the data matrix should be 2904, that is 484×6, but some companies don't have available data for all the years.

normalized between 0 and 1 (in the case of the data being integer, boolean or real positive) or between -1 and 1 otherwise.

One of the problems with this database is that some of the data are missing. Specifically, around 16% of the companies in the database have one or more data values missing. To handle this we have substituted the missing data for the minimum value that variable can take. After the normalization the value will be set to 0 or -1 depending whether the variable can take negative values or not.

2.1 Training and Testing Sets

In order to apply GP to the prediction problem the data sets have been divided into two groups: the training and testing sets, which have been selected randomly. Given that the data base is highly unbalanced (only 5-6% of the companies went bankrupt), this ratio needs to be reflected in the choice of the training set.

The number of companies with available data varies slightly each year. In year 1999 there are data available from 467 companies (27 bankrupt vs. 440 healthy) and in year 2000 there are data available from 479 companies (30 bankrupt vs. 449 healthy). We have kept constant the number of companies in the training set, thus the number of companies in the testing set varies from year to year.

The division of the data into the training and testing sets has been done as follows. The training set consist of 160 companies (10 bankrupt and 150 healthy). The test set for year 1999 consists of 307 companies (17 bankrupt and 290 healthy) and the test set for year 2000 consists of 319 companies (20 bankrupt and 299 healthy).

3 Genetic Programming and Prediction

In this section we briefly describe the GP framework that we have used for representing systems for bankruptcy prediction. Basically, the GP algorithm must find a structure (a function) which can, once supplied with the relevant data from the company, decide if this company is heading for bankruptcy or not. In short, it is a 2-class classification problem for GP to solve. One of the classes consists of the companies that will go bankrupt and the other consists of the healthy ones. For further information on classification using GP see references [8,12].

3.1 Function and Terminal Sets

Prior to creating a GP environment the designer must define which functions (internal tree nodes) and terminals (leaf branches) are relevant for the problem to solve. This choice defines the search space for the problem in question.

The terminal set consists of 30 company data. These data are presented to the classifier as a vector and, in order to simplify the notation, they have been called $x_0, x_1 ... x_{29}$. The evolution process will decide which of these data are relevant for the solution of the problem.

Table 1. Function and terminal set

Nodes	No. arguments	Description
$\mathcal{R}$	0	random constant
$x_0 \ldots x_{29}$	0	company data
cos, *log*, *exp*	1	cosine, logarithm, exponential
$+, -, *, /$	2	arithmetic operators
IfLTE	4	if $arg_1 \leq arg_2$ then arg_3 else arg_4

Table 1 shows the function and terminal sets used. Some of these functions, such as the division, the exponential and the logarithm, have been implemented with a protection mechanism to avoid incomputable results (e. g. division by zero returns zero and the logarithm returns the logarithm of the absolute value of its argument, or 0 if the argument is 0).

3.2 Classification

The classification works as follows. Let $X = \{x_0, \ldots, x_{\mathrm{N}}\}$ be the vector comprising the data of the company undergoing classification. Let $f(X)$ be the function defined by an individual GP tree structure. The value y returned by $f(X)$ depends on the input vector X.

$$y = f(x_0, x_1, \ldots, x_{\mathrm{N}}) \tag{1}$$

We can apply X as the input to the GP tree and calculate the output y. Once the numerical value of y has been calculated, it will give us the classification result according to:

$$y > 0,\ \ X \in B \tag{2}$$

$$y \leq 0,\ \ X \in \overline{B} \tag{3}$$

where B represents the class to which bankrupt companies belong and $\overline{B}$ represents the class to which healthy companies belong.

That is, if the evaluation of the GP tree results in a numerical value greater than 0 the company is classified as heading for bankruptcy, while if the value is less or equal to 0 the company is classified as healthy.

3.3 Fitness Evaluation

As mentioned previously, the database we are using is very unbalanced in the sense that only 5-6% of the companies included will go bankrupt. This is something to take into account while designing the fitness function. Otherwise the evolution may converge to structures that classify all companies as healthy (i.e. they do not classify at all) and still get a 95% hits rate. According to [10] there are 3 ways to address this problem:

- Undersampling the over-sized class
- Oversampling the small class
- Modifying the cost associated to misclassifying the positive and the negative class to compensate for the imbalanced ratio of the two classes. For example, if the imbalance ratio is 1:10 in favour of the negative class, the penalty of misclassifying a positive example should be 10 times greater.

We have used the cost-modifying approach, not only because it was the one recommended by the authors of [10] but mainly because the oversampling and undersampling approaches did not yield good results.

Therefore, the fitness function is:

$$Fitness = \sum_{i=1}^{n} u_i \tag{4}$$

where

$$u = \begin{cases} 0 & : \text{ incorrect classification} \\ 1 & : \text{ bankrupt company classified correctly} \\ \frac{n_{b=0}}{n_{b=1}} & : \text{ healthy company classified correctly} \end{cases} \tag{5}$$

$n_{b=0}$ is the number of bankrupt companies in the training set and $n_{b=1}$ is the number of healthy companies in the training set.

3.4 GP Algorithm

The GP implementation used is based on the JEO (Java Evolving Objects) library [3] developed within the project DREAM (Distributed Resource Evolutionary Algorithm Machine) [2]. The project's aim was to develop a complete distributed peer-to-peer environment for running evolutionary optimization applications over a set of heterogeneous distributed computers.

JEO is a software package in Java integrated in DREAM. In the context of GP, JEO includes a tree-shaped genome structure and several operators, therefore the user only needs to implement the methods that are problem dependent, i. e. fitness evaluation and construction of the function and terminal sets.

As a method of bloat control we have included a new crossover operator, bloat-control-crossover, that occurs with a probability of 0.45. This crossover operator implements a bloat control approach described in [9] and inspired in the "prune and plant" strategy used in agriculture. It is used mainly for fruit trees and it consist of pruning some branches of trees and planting them in order to grow new trees. The idea is that the worst tree in a population will be substituted by branches "pruned" from one of the best trees and "planted" in its place. This way the offspring trees will be of smaller size than the ancestors, effectively reducing bloat.

Table 2 shows the main parameters used during evolution.

Table 2. GP parameters

Initialization method	Ramped half and half
Replacement operator	Generational with elitism (0.2%)
Selection operator	Tournament selection
Tournament group size	10
Cloning rate	0.05
Crossover operator	Bias tree crossover
Internal node selection rate	0.9
Crossover rate	0.5
Crossover for "bloat" control rate	0.45
Tree maximum initial depth	7
Tree maximum depth	18
Population size	500
Number of runs	20
Termination criterion	50 generations

4 Results

The results obtained for each year are shown in the following tables. The first row of the tables shows the best result obtained and the second row shows the averaged results over 20 runs. Each table shows the results obtained in training, in testing and overall (i.e. training+testing). The first column shows the percentage of hits scored (i.e. the number of correct predictions), the second column presents the percentage of true positives (TP) (i.e. the number of companies heading for bankruptcy classified correctly) and the third column presents the percentage of true negatives (TN) (i.e. the number of healthy companies classified as such).

In the problem of bankruptcy prediction the results are very linked to the database in use so it is important to present results obtained with an alternative classification method for comparison purposes.

This section includes an alternative set of results obtained using a support vector machine (SVM). In order to generate these results we have used LIBSVM [6], an integrated software for support vector classification. We have chosen this particular software because it supports weighted SVM for unbalanced data. The SVM has been run 20 times using the same random training and testing sets as in the GP case.

4.1 Prediction of Bankruptcy in 2003 Using Data from Year 1999

Table 3 shows that the results obtained with GP in the training are very good. However, the average results in the testing show an average percentage of true positives smaller than 50%. This is due to GP converging to structures that achieve a good global result but at the expense of a very low percentage of true positives (i.e. the structure is classifying all companies as healthy). This explains why the average percentage of hits is greater than the one obtained by the best GP tree. Nevertheless, it can be seen that the best GP achieves

Table 3. GP results using data from year 1999

	Training			Testing			Overall		
	% hits	% TP	% TN	% hits	% TP	% TN	% hits	% TP	% TN
Best GP results	88.12	90.00	88.00	74.92	76.47	74.83	79.44	81.48	79.32
Avg. GP results	89.69	98.50	89.10	80.94	40.59	83.31	83.94	62.04	85.28

very satisfactory and balanced percentages of hits in the classification of both bankrupt and healthy companies.

The best GP individual can be expressed as follows:

$$\begin{aligned}
y &= x_{29}\cos(40.97 - x_{28} + \exp(t_2))t_3 \qquad (6)\\
t_1 &= \exp(x_9 x_{29})\\
t_2 &= \cos(t_7 x_{29} + t_6(x_{29} + x_{20} + 40.97 - x_{28} + t_1))\\
t_3 &= t_4 + x_{29}(\cos(2x_{23}) + \cos(x_{29}(40.97 - x_{28}) + t_1) + 81.94 + t_1 - x_{28} + t_5\\
t_4 &= \exp(\exp(\cos(\cos(\exp(x_7) + x_{20}))\cos(40.97 - x_{28} + x_{23} + x_{20})))\\
t_5 &= x_{20}\cos(t_1 + x_{20}) + \cos(40.97 - x_{28} + x_{26}) + \cos(x_{29}(t_6 + t_7) + x_{15})\\
t_6 &= \cos(\exp(\exp x_7) + x_{20})\\
t_7 &= \cos(\exp(\exp x_7) + x_{23})
\end{aligned}$$

When compared with the results obtained with SVM (see table 4), it can be seen that the GP results are superior.

Table 4. SVM results using data from year 1999

	Training			Testing			Overall		
	% hits	% TP	% TN	% hits	% TP	% TN	% hits	% TP	% TN
Best SVM results	64.37	90.00	62.67	71.01	94.12	69.66	68.74	92.59	62.27
Avg. SVM results	70.44	96.50	68.70	67.95	88.53	66.74	68.80	91.48	67.41

The percentage of TP is slightly better for the SVM, but the overall performance of the GP is better (79.44% versus 68.74% of hits) and more balanced. To further check if the difference in results between GP and SVM is real we have performed a Mann-Whitney U test to the testing results. The test has concluded that the results are statistically different.

Table 5. GP results using data from year 2000

	Training			Testing			Overall		
	% hits	% TP	% TN	% hits	% TP	% TN	% hits	% TP	% TN
Best GP results	77.50	100.00	76.00	73.04	80.00	72.57	74.53	86.67	73.72
Avg. GP results	89.81	99.50	89.17	79.94	43.75	82.36	83.24	62.33	84.63

4.2 Prediction of Bankruptcy in 2003 Using Data from Year 2000

The results obtained with GP using data from year 2000 are shown in table 5. The average results show that, due to the unbalanced data, there are some GP structures that do not achieve good percentages of true positives. The percentages of hits obtained with the best GP result are very satisfactory.

The best GP individual consists of 12 nested conditional clauses:

$$\begin{aligned}
y &= \text{ if } x_{28} \le x_{11} \text{ then } (f_0 - f_1)/x_{29}^2 \text{ else } f_2 \qquad (7)\\
f_0 &= \text{ if } x_{26} \le (f_4 - f_3) \text{ then } x_{29} \text{ else } x_{17}\\
f_1 &= \text{ if } x_{19} \le x_4 \text{ then } x_{26} \text{ else } f_7\\
f_2 &= \text{ if } f_8 \le (x_{19} - x_{12} - x_3) \text{ then } x_{29} \text{ else } x_{17}\\
f_3 &= \text{ if } x_{19} \le x_4 \text{ then } x_{26} \text{ else } f_5\\
f_4 &= \text{ if } x_{26} \le (f_6 - f_3) \text{ then } x_{29} \text{ else } x_{17}\\
f_5 &= \text{ if } x_{12} \le x_{10} \text{ then } x_{28} \text{ else } x_5\\
f_6 &= \text{ if } x_{26} \le -4x_{12} \text{ then } x_{29} \text{ else } x_{17}\\
f_7 &= \text{ if } x_{12} \le x_{27} \text{ then } x_{28} \text{ else } x_5\\
f_8 &= \text{ if } x_{17} \le x_{11} \text{ then } f_5/x_{29}^2 \text{ else } f_9\\
f_9 &= \text{ if } f_5 \le f_{10} \text{ then } x_{29} \text{ else } x_{17}\\
f_{10} &= \text{ if } x_{26} \le (x_{18} - x_3 - 3x_{12}) \text{ then } x_{29} \text{ else } x_{17}
\end{aligned}$$

Again the overall performance of the GP is better (74.53% versus 69.10% of hits) and more balanced (see table 6). The Mann-Whitney U test has also confirm that the results obtained with GP and SVM are statistically different.

Table 6. SVM results using data from year 2000

	Training			Testing			Overall		
	% hits	% TP	% TN	% hits	% TP	% TN	% hits	% TP	% TN
Best SVM results	69.37	90.00	68.00	68.97	95.00	67.22	69.10	93.33	67.48
Avg. SVM results	69.28	98.50	67.33	67.29	87.25	65.95	67.95	91.00	66.41

5 Conclusions

The main problems we had to handle in this work was the imbalance between the number of companies heading for bankruptcy (around 5-6%) and the number of healthy companies, and the amount of missing data (around 16% of the companies have one or more missing data) in the database we have used for the analysis.

The approaches we have used to solve them have been the normalization of the data and the use of a fitness function that suited the unbalance problem.

The results obtained are very satisfactory. The best GP structure has achieved a percentage of hits of around 75% in the testing set. When compared with the numerical results obtained with SVM they are clearly better. In addition GP provides us with a nonlinear function in a tree shape, which is easier to analyze and draw conclusions from than the SVM black box structure.

In the future we plan to combine data from several years to carry out the prediction. This raises the problem of how to handle the data. We plan to use serial processing of data. Instead of presenting the data to the system as a vector (i.e. simultaneously) we will adopt an alternative approach in which the data from various years is presented to the classifier in series.

Acknowledgments

Thanks to Isabel Román Martínez, José M. de la Torre Martínez y Elena Gómez Miranda, from the Department of Finance and Accounting of the Universidad de Granada, for the financial database. This work was supported by the Plan Nacional de I+D of the Spanish Ministerio de Educación y Ciencia (NADEWeb project - TIC2003-09481-C04).

References

1. Altman, E. I.: Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J. of Finance **23 (4)** (1968) 589–609
2. Arenas, M. G., Collet, P.,Eiben, A. E., Jelasity, M., Merelo, J. J., Paechter, B., Preuß, M., Schoenauer, M.: “A framework for distributed evolutionary algorithms”. Proc. of PPSN’02, in LNCS (Springer-Verlag) **2439** (2002) 665–675
3. Arenas, M. G., Dolin, B., Merelo, J. J., Castillo, P. A., Férnandez de Viana, I., Schoenauer, M.: “JEO: Java Evolving Objects”. Proc. of the Genetic and Evolutionary Computation Conf., GECCO’02. New York, USA (2002) 991–994
4. Brabazon, A., Keenan, P. B.: A hybrid genetic model for the prediction of corporate failure. Computational Management Science **1** (2004) 293–310
5. Brabazon, A., O’Neill, M.: Biologically inspired algorithms for finantial modelling. Springer-Verlag, Berlin, 2006
6. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (2001)
7. Dimitras, A. I., Zanakis, S. H., Zopounidis, C.: A survey of business failures with an emphasis on predictions, methods and industrial applications. Eur. J. Oper. Res. **90** (1996) 487–513

8. Eggermont, J., Eiben, A. E., van Hemert, J. I.: "A comparison of genetic programming variants for data classification". Proc. of IDAC'99, in LNCS (Springer-Verlag) **1642** (1999) 281–290
9. Fernández de Vega, F., Rubio del Solar, M., Fernández Martínez, A.: "Plantación de árboles: Una nueva propuesta para reducir esfuerzo en programación genética". Actas del IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB'05. Granada, Spain (2005) 57–62.
10. Japkowicz, N., Stephen, S.: The class imbalance problem: a systematic study. Intelligent Data Analysis **6:5** (2002) 429-449
11. Kim, M. J., Han, I.: The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. Expert Syst. Appl. **25** (2003) 637–646
12. Kishore, J. K., Patnaik, L. M., Mani, V., Agrawal, V. K.: Genetic programming based pattern classification with feature space partitioning. Inf. Sci. **131** (2001) 65–86
13. Lensberg, T., Eilifsen, A., McKee, T. E.: Bankruptcy theory development and classification via genetic programming. Eur. J. Oper. Res. **169** (2006) 677-697
14. Leshno, M., Spector, Y.: Neural network predction analysis: The bankruptcy case. Neurocomputing **10** (1996) 125–147
15. McKee, T. E., Lensberg, T.: Genetic programming and rough sets: A hybrid approach to bankruptcy classification. Eur. J. Oper. Res. **138** (2002) 436–451
16. Ohlson, J.: Financial ratios and the probabilistic prediction of bankruptcy. J. of Accounting Research **18 (1)** (1980) 109–131
17. Salcedo-Sanz, S., Fernández-Villacañas, J. L., Segovia-Vargas, M. J. Bousoño-Calzón, C.: Genetic programming for the prediction of insolvency in non-life insurance companies. Computers & Operations Research **32** (2005) 749–765
18. Shin, K. S., Lee, Y. L.: A genetic algorithm application in bankruptcy prediction modeling. Expert Syst. Appl. **23** (2002) 321–328
19. Tsakonas, A., Dounias, G., Doumpos, M., Zopounidis, C.: Bankruptcy prediction with neural logic networks by means of grammar-guided genetic programming. Expert Syst. Appl. **30** (2006) 449-461
20. Varetto, F.: Genetic algorithm applications in the field of insolvency risk. Journal of Banking and Finance **22** (1998) 1421–1439
21. Vieira, A. S., Ribeiro, B., Mukkamala, S., Neves, J. C., Sung, A. H.: "On the performance of learning machines for bankruptcy detection". Proc. of the IEEE Conf. on Computational Cybernetics. Vienna, Austria (2004) 323–327