Corporate Failure Prediction Modeling Using Genetic Algorithm Technique

Kyung-shik Shin and Ingoo Han
Graduate School of Management
Korea Advanced Institute of Science & Technology

ABSTRACT

We propose a genetic algorithms (GAs) approach in this study and illustrate how GAs can be applied to corporate failure prediction modeling. Although numerous theoretical and experimental studies reported the usefulness of neural networks in classification studies, there exists a major drawback in building and using the model. That is, the user can not readily comprehend the final rules that the neural network models acquire. An advantage of GAs approach offers is that they are capable of extract rules that are easy to understand for users like expert systems. The preliminary results show that rule extraction approach using GAs for bankruptcy prediction modeling is promising.

I. INTRODUCTION

Today, Korean financial institutions are paying a heavy price for their indiscriminate practices. Corporate bankruptcies have put several institutions on the brink of insolvency. Many others will be merged with or acquired by other financial institutions. Surviving institutions are rushing to put in place a corporate credit rating system, but are facing difficulties due to a lack of data accumulation and scientific credit rating methods.

The present research pertains to a corporate failure prediction modeling which can provide a basis for credit rating system. Prediction of corporate failure using past financial data is also a well-documented topic. Early studies of bankruptcy prediction used statistical techniques such as multiple discriminant analysis (Altman, 1968, 1983), logit (Ohlson, 1980), probit (Zmijewski, 1984). Recently, however, numerous studies have demonstrated that artificial intelligence such as neural networks (NNs) can be an alternative methodology for classification problems to which traditional statistical method have long been applied (Barniv *et al.*, 1997; Bell, 1997; Boritz and Kennedy, 1995; Chung and Tam, 1992; Etheridge and Sriram, 1997; Fletcher and Goss, 1993; Jo *et al.*, 1997; Odom and Sharda, 1990; Salchenberger *et al.*, 1992; Shin and Han, 1998a; Shin *et al.*, 1998; Tam and Kiang, 1992; Wilson and Sharda, 1994).

Although numerous theoretical and experimental studies reported the usefulness of NNs in classification studies, there are several drawbacks in building and using the model. First, it is an art to find an appropriate NN model which can reflect problem characteristics because there are numerous network architectures, learning methods, and parameters. Second, the user can not readily comprehend the final rules that the neural network models acquire. This characteristic of NNs is often referred to 'Black boxes'.

We propose a genetic algorithms (GAs) approach in this study and illustrate how GAs can be applied to corporate failure prediction modeling. An advantage of this approach offers is that they are capable of extract rules that are easy to understand for users like expert systems.

The remainder of this paper is organized as follows. The second section provides a brief description of GAs. The third section describes the rule extraction approach using genetic search. The fourth section reports the model development and the results of experiments. The final section discusses the conclusions and future research issues.

II. GENETIC ALGORITHM TECHNIQUE

GAs are stochastic search techniques that can search large and complicated spaces on the ideas from natural genetics and evolutionary principle (Davis,1991; Holland, 1975; Goldberg, 1989). They have been demonstrated to be effective and robust in searching very large spaces in a wide range of applications (Colin, 1994; Han, Jo & Shin, 1997; Klimasauskas, 1992; Koza, 1993). GAs are particularly suitable for multi-parameter optimization problems with an objective function subject to numerous hard and soft constraints.

The financial application of GAs is growing with successful applications in trading system (Colin, 1994; Deboeck, 1994), stock selection (Mahfoud and Mani, 1995), portfolio selection (Rutan, 1993), bankruptcy prediction (Kingdom and Feldman, 1995), credit evaluation (Shin and Han, 1998b; Walker *et. al.*, 1995) and budget allocation (Packard, 1990).

GAs perform the search process in four stage: initialization, selection, crossover, and mutation (Davis, 1991; Wong & Tan, 1994). Figure 1 shows the basic steps of genetic algorithms. In the initialization stage, a population of genetic structures (called chromosomes) that are randomly distributed in the solution space, is selected as the starting point of the search. After the initialization stage, each chromosome is evaluated using a user-defined fitness function. The goal of the fitness function is to numerically encode the performance of the chromosome. For real-world applications of optimization methods such as GAs, the choice of the fitness function is the most critical step.

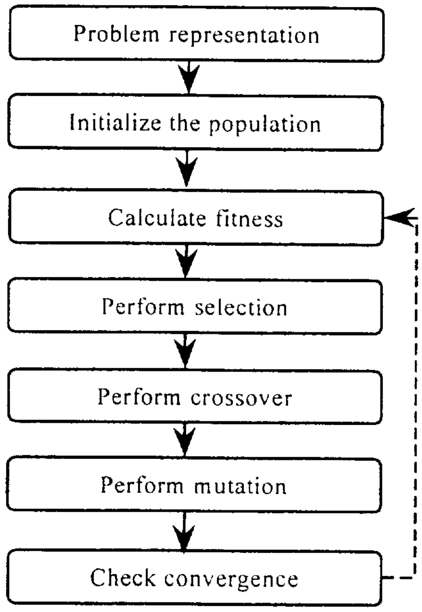

Figure 1. Basic steps of genetic algorithms

The mating convention for reproduction is such that only the high scoring members will preserve and propagate their worthy characteristics from generations to generation and thereby help in continuing the search for an optimal solution. The chromosomes with high performance may be chosen for replication several times whereas poor-performing structures may not be chosen at all. Such a selective process causes the best-performing chromosomes in the population to occupy an increasingly larger proportion of the population over time.

Crossover causes to form a new offspring between two randomly selected 'good parents'. Crossover operates by swapping corresponding segments of a string representation of the parents and extends the search for new solution in far-reaching direction. The crossover occurs only with some probability (the crossover rate). There are many different types of crossover that can be performed: the one-point, the two-point, and the uniform type (Syswerda,1989).

Mutation is a GA mechanism where we randomly choose a member of the population and change one randomly chosen bit in its bit string representation. Although the reproduction and crossover produce many new strings, they do not introduce any new information into the population at the bit level. If the mutant member is feasible, it replaces the member which was mutated in the population. The presence of mutation ensures that the probability of reaching any point in the search space is never zero.

III. RULE EXTRACTION USING GAs

In building corporate failure predicting model, we use the similar approach that Bauer (1994), and Mahfoud and Mani (1995) suggest in their stock selection applications. We apply GAs to find thresholds (cutoffs) for one or more variables, above or below which a company is considered 'dangerous'. For instance, if the model's structure consists of two variables representing a particular company's quick ratio and a debt ratio, the final rule the GA returns might look like the following:

IF [Debt ratio > 1.50 and Quick ratio < 0.35] THEN Dangerous

In many cases, the simplistic rule like above example is insufficient to model relationships among financial variables. Our rule structure contains five conditions using 'AND' relations for this study. The general form of rule that GAs generate is as follows:

IF [the VAR1 is GREATER THAN OR EQUAL TO (LESS THAN) X1,
AND the VAR2 is GREATER THAN OR EQUAL TO (LESS THAN) X2,
AND ....
AND the VAR5 is GREATER THAN OR EQUAL TO (LESS THAN) X5]
THEN Prediction is Dangerous.

If the all of five conditions are satisfied, then the model will produce 'dangerous' signal on evaluating company. X1 to X5 denotes the cutoff values which are found through genetic search process. The cutoff values range from 0 to 1, and represent the percentage of the data source's range. This allows the rules to refer to any data source, regardless of the values it takes on. Above rule structure is summarized in Table 1. In the table, 'which data' means data source the rule refers to. The general rule structure is illustrated in Table 1.

Table 1. The general rule structure

Number(j)	Cond1	Cond2	Cond3	Cond4	Cond5	Description
Which data	VAR_{1i}	VAR_{2i}	VAR_{3i}	VAR_{4i}	VAR_{5i}	VAR_{ji} (i=var. number, j = cond. number)
Less than / greater than or equal to	L/G_{1k}	L/G_{2k}	L/G_{3k}	L/G_{4k}	L/G_{5k}	L/G_{jk} (k= 1: less than / 2: greater than or equal to)
Cutoff values	X_1	X_2	X_3	X_4	X_5	Cutoff X_j (j = cond. number)

We allow the model to select 5 variables among 9 alternative financial ratios. We also allow to choosing one variable more than once in the rule structure because the bankruptcy prediction is often highly nonlinear task. When a 10% increase in sales results is a good signal, however, a 100% increase in that same variable could result in a bad signal. This means that use of multiple cutoff points if necessary to extract knowledge from financial variables is recommended. In addition, if we consider the interaction of conditions, this flexibility is essential to financial modeling.

In setting up the genetic optimization problem, we need the parameters that have to be coded for the problem and an objective or fitness function to evaluate the performance of each string. The parameters that are coded are the cell values of Table 1. As we mentioned above, they are input variables, above or below, and the cutoff values. The varying parameters generate a number of combinations of our general rules. The string encoded for the experiments is as follows:

$$\text{String } \{ VAR_{1i}, VAR_{2i}, VAR_{3i}, VAR_{4i}, VAR_{5i}, L/G_{1k}, L/G_{2k}, L/G_{3k}, L/G_{4k}, L/G_{5k}, X_1, X_2, X_3, X_4, X_5 \}$$

The GAs maintain a population of strings which are chosen at random. This initialization allows the GAs to explore the range of all possible solutions, and this tends to favor the most likely solutions. Generally, the population size is determined according to the size of the problem (bigger population for larger problem). The common view is that a larger population takes longer to settle on a solution, but is more likely to find a global optimum because of its more diverse gene pool. We use 100 strings in the population.

The task of defining a fitness function is always application specific. In this study, the objective of the system is to find a rule which would yields the highest hit ratio if rules are fired across the company. We apply the hit ratio of the rule to the fitness function for this study.

The genetic operators such as crossover and mutation which are described in the previous section are used to search for the optimal solutions. Several parameters must be defined for the above operators, and the values of these parameters can greatly influence the performance of the algorithm. The crossover rate ranges 0.5 - 0.7 and the mutation rate ranges 0.06 - 0.12 for our experiment. As a stopping condition, we use 3,000 trials. These processes are done by the genetic algorithms software package Evolver(TM) 4.0, called from an Excel macro.

IV. EXPERIMENT AND RESULTS

4.1 Data and Variables

The data set contains 528 mid-size manufacturing firms (externally audited firms) which filed for bankruptcy (264 cases) and non-bankruptcy (264 cases) during the period 1995-1997. We apply two stages of input variable selection process. In the first stage, we select 55 variables by factor analysis, independent-samples t-test (between input variable and output variable) and Mann-Whitney U test (for qualitative variables). In the second stage, we select 9 financial variables using the stepwise methods to reduce the dimensionality. The aim of input variable selection approach is to select the input variables satisfying the univariate test first, and then select significant variables by stepwise method for refinement. As we mentioned above, these variables are not the final ones that are used to form a rule, but are provided as the alternative variables for the final selection. Table 2 illustrates the pre-selected variables for this study.

Table 2. Selected variables

Variables	Name
X1	Value added to total asset
X2	Net income to stockholder's equity
X3	Quick ratio
X4	Liquidity ratio
X5	Current liability to total assets
X6	Retained earnings to total assets
X7	Stockholders' equity to total assets
X8	Financial expenses to sales
X9	Operating income to operating expenses

Each data set is split into two subsets, a training set and a validation (holdout) set of 85 and 15 percent of data, respectively. The training data are used for learning rules, and the validation data are used to test the results with the data which have not been used to develop the system.

4.2 Results

We extract five bankruptcy rules by genetic search process. The rules generated and corresponding descriptions are illustrated in Table 3 and Table 4.

The goal in optimization is ideally to find the best solution to a problem. Since GAs try to find out the optimal or near optimal combination of above searching parameters, the final solution is one. However, in real-word problem solving, one does not usually know the best possible solution. Therefore, a more realistic objective is to find alternatively good solutions. We generate multiple rules by choosing multiple strings in the converged population. Since the fitness function of GAs measures the quality of a particular solution, we select the strings which show high level of fitness values. So the derived rules are alternatively good rules which show high level of hit ratio although there are minor differences in simulated performance.

Table 3. The rules generated

Rule number		Cond 1	Cond 2	Cond 3	Cond 4	Cond 5
Rule 1	Variable code	2	4	5	7	8
	> / < code	1	1	1	1	1
	Cutoffs	0.426	0.847	0.520	0.595	0.665
Rule 2	Variable code	2	2	3	7	8
	> / < code	1	1	1	1	1
	Cutoffs	0.520	0.595	0.697	0.590	0.503
Rule 3	Variable code	2	4	6	7	8
	> / < code	1	1	2	1	1
	Cutoffs	0.426	0.560	0.082	0.590	0.520
Rule 4	Variable code	2	3	6	7	8
	> / < code	1	1	2	1	1
	Cutoffs	0.560	0.697	0.130	0.577	0.515
Rule 5	Variable code	2	3	6	7	8
	> / < code	1	1	2	1	1
	Cutoffs	0.560	0.697	0.082	0.590	0.520

Table 4. The description of rules

Rule number	Description
Rule 1	IF Net income to stockholder's equity is less than 0.426* AND Liquidity ratio is less than 0.847 AND Current liability to total assets is less than 0.520 AND Stockholders' equity to total assets is less than 0.595 AND Financial expenses to sales is less than 0.665, THEN Dangerous.
Rule 2	IF Net income to stockholder's equity is less than 0.520 AND Quick ratio is less than 0.697 AND Stockholders' equity to total assets is less than 0.590 AND Financial expenses to sales is less than 0.503, THEN Dangerous.
Rule 3	IF Net income to stockholder's equity is less than 0.426 AND Liquidity ratio is less than 0.560 AND Retained earnings to total assets is greater than or equal to 0.082 AND Stockholders' equity to total assets is less than 0.590 AND Financial expenses to sales is less than 0.590, THEN Dangerous.
Rule 4	IF Net income to stockholder's equity is less than 0.560 AND Quick ratio is less than 0.697 AND Retained earnings to total assets greater than or equal to 0.130 AND Stockholders' equity to total assets is less than 0.577 AND Financial expenses to sales is less than 0.515, THEN Dangerous.
Rule 5	IF Net income to stockholder's equity is less than 0.560 AND Quick ratio is less than 0.697 AND Retained earnings to total assets is greater than or equal to 0.082 AND Stockholders' equity to total assets is less than 0.590 AND Financial expenses to sales is less than 0.520, THEN Dangerous.

* Represent the percentage of the data source's range.

The hit ratios calculated from simulation results are summarized in Table 5. In table 5, hit ratio(A) denotes the rate of correct classification if the rule is fired, while hit ratio(B) represents overall classification accuracy of the set.

Table 5. The performance of derived rules (%)

Rules	Train (476 cases)		Validation (52 cases)		
	Hit ratio(A)	Hit ratio(B)	Hit ratio(A)	Hit ratio(B)	# of rules fired
Rule1	79.0	78.8	84.6	80.0	30
Rule2	80.7	80.0	76.9	75.0	28
Rule3	82.6	80.0	84.6	82.1	28
Rule4	81.6	79.6	78.9	77.8	27
Rule5	80.2	80.0	78.9	77.8	27
Average	80.8	79.7	80.8	78.5	28

The average hit ratio if the rules are fired is 80.8% of training and validation sets, repectively. This means if the financial variables of a company are within the feature ranges of derived rules, the probability of bankruptcy is about 80% of cases.

The preliminary results above demonstrate that GAs are effective methods for extracting rules for the bankruptcy prediction. Their success is due to their ability to learn nonlinear relationships among the input variables. A drawback of this approach is that the model produces predictions only when the rules are fired. while NNs make predictions on every case except when explicitly restricted. The average number of cases that are fired by a specific rule is 28 among 52 cases (53.8%). This problem, however, can be reduced by integrating multiple rules derived. We have many ways to integrate these rules. For example, if one of the five rules makes 'Danger' signal, the model may produce 'Danger' signal to the users.

CONCLUDING REMARKS

We applied GAs to extract rules that can predict corporate failure. This paper is just a first attempt to explore the potential of genetic-based systems to handle bankruptcy prediction problems systematically. The results show that rule extraction approach using GAs for bankruptcy prediction modeling is promising.

This paper has several limitations. First, although we derived multiple rules using traditional GAs, it is necessary to extend the GAs through use of a niching method (Mahfoud and Mani, 1995). Unlike the traditional GAs, which makes the population eventually converge around a single point in the solution space, the GA that uses a niching method converges about multiple solution or niches.

Second, the current rule structure is quite limited. As a next research step, this structure will be considerably extended by incorporating additional features. It is likely that more informative features will possibly leading to improved results, although we should consider the efficiency problem. Further improvements may be obtained by incorporating qualitative factors and quantitative ones. We plan to include qualitative variables in extracting the prediction rules.

REFERENCES

Altman, E. (1968), "Financial ratios, discriminant analysis and the prediction of corporate bankruptcy," *The Journal of Finance*, 23, pp.589-609.

Altman, E. (1983), *Corporate financial distress – A complete guide to predicting, avoiding and dealing with bankruptcy*, New York: John Wiley.

Barniv, R., Agarwal, A. and Leach, R. (1997), "Predicting the outcome following bankruptcy filing: A three-state classification using neural networks," *Intelligent Systems in Accounting, Finance and Management*, 6, pp.177-194.

Bauer, R. J. (1994), *Genetic Algorithms and Investment Strategies*, John Wiley & Sons.

Bell, T. (1997), "Neural nets or the logit model? A comparison of each model's ability to predict commercial bank failures," *Intelligent Systems in Accounting, Finance and Management*, 6, pp.249-264.

Boritz, J. and Kennedy, D. (1995), "Effectiveness of neural networks types for prediction of business failure," *Expert Systems with Applications*, 9, pp.503-512.

Chung, H. and Tam, K. (1992), "A Comparative Analysis of Inductive Learning Algorithm". *Intelligent Systems in Accounting, Finance and Management*, vol. 2, pp.3-18.

Colin, A. M., (1994), "Genetic algorithms for financial modeling," In Deboeck, G.J. (Eds.), *Trading On The Edge*, New York: John Wiley, pp.148-173.

Davis, L. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, NY.

Deboeck, G. J., (1994), "Using GAs to optimize a trading system," In Deboeck, G.J (Eds.), *Trading On The Edge*. New York: John Wiley, pp.174-188.

Etheridge, H. and Sriram, R. (1997), "A comparison of the relative costs of financial distress models: Artificial neural networks, logit and multivariate discriminant analysis," *Intelligent Systems in Accounting, Finance and Management*, 6, pp.235-248.

Fletcher, D. and Goss, E. (1993), "Forecasting with neural networks: An application using bankruptcy data," *Information and Management*, 24(3), pp.159-167.

Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, MA: Addison-Wesley.

Han, I., Jo, H. and Shin, K. S. (1997), "The hybrid systems for credit rating," *Journal of the Korean Operations Research and Management Science Society*, 22(3), pp.163-173.

Holland, J. H. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor, The University of Michigan Press.

Jo, H., Han, I. And Lee, H. (1997), "Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis, " *Expert Systems With Applications*, 13 (2), pp.97-108.

Kingdom, J. and Feldman, K. (1995), *Genetic algorithms for bankruptcy prediction*, Search Space Research Report, No.01-95, Search Space Ltd. London.

Klimasauskas, C. C. (1992), "Hybrid neuro-genetic approach to trading algorithms," *Advanced Technology for Developers*, 1(7).

Koza, J., (1993), *Genetic programming*, Cambridge: The MIT Press.

Mahfoud, S. and Mani, G. (1995), "Genetic algorithms for predicting individual stock performance," *Proceedings of the 3rd International Conference on Artificial Intelligence Applications on Wall Street*, pp.174-181.

Odom, M. and Sharda, R. (1990), "A neural networks model for bankruptcy prediction," *Proceedings of the IEEE International Conference on Neural Network*, 2, pp.163-168.

Ohlson, J. (1980), "Financial ratios and the probabilistic prediction of bankruptcy," *Journal of Accounting Research*, 18(1), pp.109-131.

Packard, N. (1990), "A genetic learning algorithm for the analysis of complex data," *Complex Systems*, 4, pp.543-572.

Rutan, E. (1993), "Experiments with optimal stock screens," *Proceedings of the 3rd International Conference on Artificial Intelligence Applications on Wall Street*, pp.269-273.

Salchenberger, L., Cinar, E. and Lash, N. (1992), "Neural networks: A new tool for predicting thrift failures," *Decision Sciences*, 23, pp.899-916.

Shin, K. S. and Han, I. (1998a), "Bankruptcy Prediction Modeling Using Multiple Neural Networks Models," *Proceedings of Korea Management Science Institute Conference*, 1998.

Shin, K. S. and Han, I. (1998b), "Using Genetic Algorithm to Support Case-Based Reasoning: Application to Corporate Bond Rating Integration," *Proceedings of Second Asia Pacific Decision Sciences Institute (DSI) Conference*, Taipei.

Shin, K. S., Shin, T. S. and Han, I. (1998), "Corporate Credit Rating System Using Bankruptcy Probability Matrix", *Forthcoming Conference Proceedings of IV International Meeting on Artificial Intelligence and Emerging Technologies in Accounting, Finance and Taxation*, Spain.

Syswerda, G. (1989), "Uniform crossover in genetic algorithms," In Schaffer, J.D. (Eds.), *Proc. 3rd Int.Conf. Genetic Algorithms*. San Maeto, CA: Morgan Kaufmann.

Tam, K. and Kiang, M. (1992), "Managerial applications of neural networks: the case of bank failure predictions," *Management Science*, 38(7), pp.926-947.

Walker, R., Haasdijk, E. and Gerrets, M. (1995), "Credit evaluation using a genetic algorithm," In Coonatilake, S. and Treleaven, P. (Eds.), *Intelligent Systems for Finance and Business*, Chichester, John Wiley, pp.39-59.

Wilson, R. and Sharda, R. (1994), "Bankruptcy prediction using neural networks," *Decision Support Systems*, 11(5), pp.545-557.

Wong, F. and Tan, C. (1994), "Hybrid neural, genetic and fuzzy systems," In Deboeck, G.J. (Eds.), *Trading On The Edge*. New York: John Wiley, pp.245-247.

Zmijewski, M. E. (1984), "Methodological Issues Related to the Estimated of Financial Distress Prediction Models", *Journal of Accounting Research*, 22(1), pp.59-82.