The Use of a Genetic Algorithm in Forecasting Air Carrier Financial Stress and Insolvency

Sergio Davalos, Ph.D., University of Washington-Tacoma
Richard D. Gritta, Ph.D., University of Portland
Bahram Adrangi, Ph.D., University of Portland
Jason Goodfriend, Ph.D., Bureau of Transportation Statistics

ABSTRACT

While statistical and artificial intelligence methods such as Artificial Neural Networks (ANN) have been used successfully to classify organizations in terms of solvency or insolvency, they are limited in degree of generalization either by requiring linearly separable variables, lack of knowledge of how a conclusion is reached, or lack of a consistent approach for dealing with local optimal solution whether maximum or minimum. This research explores the use of a method that has the ability of the ANN method to deal with linearly inseparable variables and incomplete, noisy data; and resolves the problem of falling into a local optimum in searching the problems space. The paper applies a genetic algorithm to a sample of U.S. airlines and utilizes financial data from carrier income statements and balance sheets and ratios calculated from this data to assess air carrier solvency.

Introduction

The forecasting of financial stress and bankruptcy in the U. S. airline industry has become quite important over the past decade, as the number of carrier insolvencies has increased steadily in the years after the deregulation of the industry. Previous research studies by the authors [Chow, Gritta, 1991; Davalos, Gritta, 2002; Goodfriend, Gritta, 2004; Gritta, Davalos, 2003} over the years have utilized different techniques such as Logit, MDA (Multiple Discriminant Analysis), and ANNs (Artificial Neural Networks). And while these statistical and artificial intelligence methods have been used successfully to classify airlines in terms of solvency or insolvency, they are limited in degree of generalization either by requiring linearly separable variables, lack of knowledge of how a conclusion is reached, or lack of a consistent approach for dealing with local optimal solution whether maximum or minimum. This research study explores the use of a method that has the ability of the ANN method to deal with linearly inseparable variables and incomplete, noisy data, and does resolve the problem of falling into a local optimum in searching the problems space.

The paper applies a genetic algorithm to a sample of U.S. airlines and utilizes financial data from carrier income statements and balance sheets and ratios calculated from this data to assess air carrier solvency. Such a study should be of interest to a numbers of groups including stockholders, bondholders, banks and other creditors, government regulators, and the flying public. It is hoped that this study will add to the research literature in this very important industry.

Overview of Genetic Algorithm

The concept behind the GA is not new. It is based on survival of the fittest and evolution. Starting with members of a candidate population of solutions, this population is evolved to the best set of solutions in an evolutionary manner. GAs are stochastic, global search techniques that can search large, complicated spaces [Goldberg, 1989]. Financial applications of GA include trading system [Colin, 1994, Deboeck, 1994], stock and portfolio selection [Mahfoud and Mani, 1995, Rutan, 1993]. Others [Back, et. al 1995] examined different methods for bankruptcy prediction. The Genetic Algorithm proved to have better performance that ANN, MDA, and logit. In their study they also examined the use of ratios or absolute numbers as the independent variables. In this research we focused on ratios that assess liquidity, leverage, activity and profitability.

The three main aspects of the Genetic Algorithm are individual coding, genetic operators, and fitness function. Solutions need to be coded into a form that can be computationally processed using the GA. Solutions can be coded as a set of bit strings such as 01001010101111 that is then converted into values. A bit can represent a presence or absence of condition or a set of bits together can represent a value for a particular attribute. This depends on the nature of the problem.

In terms of the genetic algorithms, the strings of bits contain information often called chromosomes or genotypes that are operated on by genetic operators. A complete string represents a combination of the presence or absence of all the relevant characteristics for a particular problem. A population of strings (solutions) is generated and evaluated by using a fitness function that determines the performance of a string. The best performing solutions are then modified with operators in order to improve the solutions.

The operators are reproduction and mutation. Reproduction is based on mating two strings through crossover (see figure 1). Strings are composed of n (4 in this case) variables serve as parents. The strings are randomly split at the crossover point that is between the second and third variables. Crossover can occur at more than one point. This is up to the implementer. The new strings generated contain the values for the first parent's two variables combined with the second parent's remaining variables, and vice versa. In mutation, individual bits are randomly switched (figure 2). The second bit in the first child was randomly selected and switched from a 1 to a 0, and the fourth bit in the fourth variable for the second child is randomly switched from a 1 to 0. These operators are used to find the best string or solution (or disjunct) for a problem.

Figure 1- Genetic Operation: reproduction

Current Pair:

Var. 1 (31)					**Var. 2 (15)**					**Var. 3 (8)**					**Var. 4 (4)**				
1	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
Var. 1 (13)					**Var. 2 (19)**					**Var. 3 (3)**					**Var. 4 (22)**				
0	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**

New members after crossover occurred after the second variable

Var. 1 (31)					**Var. 2 (15)**					**Var. 3 (3)**					**Var. 4 (22)**				
1	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**
Var. 1 (13)					**Var. 2 (19)**					**Var. 3 (8)**					**Var. 4 (4)**				
0	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**

Figure 2- Genetic Operation Mutation

Current Pair:

Var. 1 (31)					**Var. 2 (15)**					**Var. 3 (8)**					**Var. 4 (4)**				
1	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
Var. 1 (13)					**Var. 2 (19)**					**Var. 3 (3)**					**Var. 4 (22)**				
0	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**

Mutation of Offspring (* indicates where mutation occurred)

Var. 1 (31)					**Var. 2 (15)**					**Var. 3 (3)**					**Var. 4 (22)**				
1	**1**	**1**	**1**	**1**	**0**	*****	**10**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**0**
Var. 1 (13)					**Var. 2 (19)**					**Var. 3 (8)**					**Var. 4 (4)**				
0	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**1**	*****	**0**

In essence, the genetic algorithm begins with a disjunct from the recursive-partitioning algorithm and always retains the best disjunct for each generation of strings. The final disjunct derived by the genetic algorithm is at least as good as the best disjunct it started with [Sikora and Shaw, 1994]. Because of the improvement in performance that can happen in a generation, the final disjunct's fitness should be better than the best rule set provided by the recursive-partitioning algorithm.

The steps in the design of the genetic algorithm are as follows:

1. A population of candidate solutions is randomly generated.
2. Members of this population are then evaluated for fitness based on a fitness function. The fitness function is used to determine how well the solution performs. The fitness function cannot be derived automatically but must be developed based on the judgment of the developer.
3. Once all members of the current population have been evaluated the next step is to remove members from the population and introduce new

members. The weakest performing members are removed. The number removed depends on how many are in the population, but this number removed must be consistent.

4. New members of the population are generated on the basis of reproduction and mutation. Through reproduction a potentially better performing offspring is produced. Through mutation new members are introduced which have the potential of introducing a solution that would not have been derived through the reproduction method since mutations are random.

5. The new population then goes through steps 2-4 until a predetermined level of fitness is reached or enough iterations have been conducted without any improvement.

Research Approach.

One approach for classification is to develop a set of rules that can be applied to population to determine classification. Genetic algorithms can be used as classifier systems consisting of a set of rules [Holland, 1992]. In this research, each rule is code as a chromosome that is divided into *n* genes, where each gene corresponds to a condition involving one attribute, and *n* is the number of predicting attributes that will be used. Genes are positional where the first gene represents the first attribute, the second gene represents the second attribute, and so on. Each i-th gene, *i=1...n*, can then be subdivided into three fields: weight (*Wi*), logical operator (*Oi*) and value (*Vi*), as shown in figure 3. Each gene then corresponds to one condition in the IF part of a rule, and the entire chromosome (individual) corresponds to the entire IF part of the rule. The THEN part is not coded into the chromosome since this incorporated into the fitness function.

Fig. 3. Representation of a chromosome (individual)

Gene 1	**…**	**…**	**Genen**
W1 O1 V1	**…**	**…**	***Wn On Vn***

The W is an integer-valued variable that indicates which field in the data set is to be used. For instance, if the data set contains the following fields: costs, revenue, taxes, profit, in that order, a 2 would indicate that we are using the second field "revenue". An O represents the logical operators "<, <=, >" and can take on the value 0, 1, 2 respectively. V is a real-valued taking on attribute values in the range [0..1]. This indicates that the data values are normalized. The following would represent the rule: If Revenue > 1000 using the above scheme: 02, 2, 1000. The commas here are used for illustrative purposes.

Each chromosome then contains several of these genes. Using the genetic algorithm various chromosome are generated and then tested using a fitness function.

Fitness Function

The fitness function evaluates the quality of each rule (individual). The fitness function is based on the following four different types of results that can occur for a prediction:

• true positive (**tp**) - the rule predicts that the firm is financially insolvent and it is.
• false positive (**fp**) **–** the rule predicts that the firm is financially insolvent and it is not.
• true negative (**tn**) - the rule predicts that the firm is financially solvent and it is;
• false negative (**fn**) - the rule predicts that that the firm is financially solvent and it is not.

Our fitness function combines two indicators commonly used in statistical analysis, namely the sensitivity (***Se***) and the specificity (***Sp***), defined as follows:

$$Se = tp / (tp + fn) \quad \textbf{(1)}$$

$$Sp = tn / (tn + fp) \quad \textbf{(2)}$$

Finally, the fitness function used by our system is defined as the product of these two indicators, i.e.:

$$fitness = Se * Sp \quad \textbf{(3)}$$

Therefore, the goal is to maximize both the ***Se*** and the ***Sp*** at the same time, and the product shown in equation (3) provides a good gradient for the function.

Research Study

For this research, 21 financial variables from the carrier income statements and balance sheets were first collected for the data set. Seven ratios were used based on the three types of financial ratios that measure liquidity, profitability, operating efficiency and financial leverage. The seven were: A liquidity measure- current liabilities to total assets (CLIAB/TA); a profitability measure- retained earnings to total assets (REARN/TA); an efficiency ratio-operating expenses to revenue (OE/REV); another profitability ratio-profit to operating expenses (PROFIT/OE); a financial leverage measure-total liabilities to total assets (TLIAB/TA); another liquidity measure- current assets to total assets (CA/TA); and lastly, another liquidity measure-current assets to operating revenue (CA/REV). These ratios where then calculated for each data point. A string of the following form was used:

String(Var1, Var2, … VarN, Op1, Op2, …, OpN, X1, X2, …, XN).

This string represented a rule that compared each of selected variable given by Var1 using the relational operator Op against the variable value, X. The particular variable used could be randomly selected. Rules were limited to only four of the seven ratios. Rules could take on the following form:

If Var1 > X1, And Var2 < X2, … And VarN > XN

then the prediction would be Solvency. (Specific operators were used to better illustrate the format).

A training set was used to train the GA and then a test set was used to evaluate the outcome. Several iterations were conducted to examine variations in performance. The average prediction accuracy was 91%. The result of the most successful GA is a rule that can then is applied to the ratios in order to determine a firm's solvency.

The Genetic Algorithm used was Solver by Palisades Systems. It is run as an Excel Add in and run on a Windows Professional system. The average turn around time was 5 minutes on a 2.2 MHZ processor. Since the data was in Excel format, the only necessary data preparation required was calculating the ratios and normalizing them.

The table below depicts the results of a run with 94% accuracy. All seven ratios were used in this run: CLIAB/TA, REARN/TA, OE/REV, PROFIT/OE, TLIAB/TASSET, CA/TA, CA/REV. The data row references the ratio used. The operators are either 1 or 2 with a 1 for "<" and a 2 for ">="

In the example below there are five rules.
Rule 1 is as follows: If CA/TA < 0.22 THEN TRUE; IF NOT THEN FALSE
Rule 2 is If CA/TA < 0.189 THEN TRUE ELSE FALSE.
Rule 3 is If CA/REV < 0.239 THEN TRUE ELSE FALSE
Rule 4 is If EXP/SALES < 0.996 THEN TRUE ELSE FALSE
Rule 5 is If TLIAB/TA < 0.603 THEN TRUE ELSE FALSE

CLIAB/TA	**REARN/TA**	**OE/REV**	**PROFIT/OE**	**TLIAB/TASSET**	**CA/TA**	**CA/REV**
1	2	3	4	5	6	7

Rule #	1	2	3	4	5	Value Range
data	**6**	**6**	**7**	**3**	**5**	1 - 7 [INT]
operator	**1**	**1**	**1**	**1**	**1**	1 - 2 [INT]
cutoff	**0.322**	**0.189**	**0.239**	**0.996**	**0.603**	0 - 1

For a company to be predicted as financially solvent, all five rules have to be true.

Conclusion

This paper has introduced a newer technique useful in forecasting air carrier financial stress and insolvency-the genetic algorithm. GAs have an advantages over statistical models, such as Logit and MDA, and artificial neural networks. While retaining the ability of the ANN method to deal with linearly inseparable variables and incomplete, noisy data, the GA resolves the problem of falling into a local optimum in searching the problems space. The GA is therefore a welcome addition to toolbox useful in forecasting stress in this important industry.

The study used data on 21 financial variables gleaned from carrier balance sheets and income statements, and ratios calculated from those variables to identify carriers that were financially stressed. The resulting GA model designed has a 91% success rate in forecasting insolvency. Future research will be aimed at increasing the reliability of the GA to predict by adding more variables to the equation and by increasing the sample size under in the design of the algorithm.

References

Barbro Back, Kaisa Sere, et al. (1995) Choosing the Best Set of Bankruptcy Predictors, *Proceedings of the 1st Nordic Workshop on Genetic Algorithms*, Vaasa, Finland, January.

Chow G., Gritta R. and Leung E. (1991). A New Approach to Forecasting Financial Distress in Air Transportation: The AIRSCORE Model. *Journal of the Transportation Research Forum*, XXXI (No. 2, Spring 1991). 371-376.

Collin (1994) Genetic Algorithms for Financial Modeling. In Debocek Guido (1994) v Trading on the Edge. John Wiley & Sons.

Davalos S., Gritta R. Wang W. (2002). Small U.S. Air Carrier Financial Condition: A Back Propagation Neural Network Approach to Forecasting Bankruptcy and Financial Stress. *Journal of the Transportation Research Forum*, LVI, No.2 (Spring 2002), 35-46.

Debocek Guido, J. (1994). Trading on the Edge. John Wiley & Sons. A Wiley Finance edition.

Fidelis, M.V., H.S. Lopes and A.A. Freitas. Discovering Comprehensible Classification Rules with a Genetic Algorithm. *Proc. Congress on Evolutionary Computation - 2000* (CEC-2000), 805-810. La Jolla, CA, USA, July/2000. IEEE, 2000.

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

Goodfriend, Jason, Gritta R., Adrangi B., and Davalos S. (2005). Assessing the Financial Condition of the Major U.S. Passenger Airlines Over the 1993-2003 Period Using the P-Score and Z" Score Discriminant Models. *Credit and Financial Management Review*, forthcoming in 2005.

Gritta R., Davalos S., and Chow G. (2003). Gauging the Financial Condition of the Major U.S. Air Carriers. *Journal of Transportation Law, Logistics and Policy*, LXXI No. 1 (Fall), 116-122.

Holland, J. H. (1992), Genetic Algorithms. *ScientificAmerican*. July 1992: 66-72

Kingdom, J. and Feldman, K.,(1995). Genetic Algorithms for Bankruptcy Prediction, *Search Space Research Report, No.01-95*, Search Space Ltd. London.

Mahfoud, S., and G. Mani. 1995. Genetic Algorithms for Predicting Individual Stock Performance. *Proceedings of the third international conference on artificial intelligence applications on Wall Street*, 174181. Gaithersburg, Md.: Software Engineering Press.

Rutan, E., (1993) Experiments with Optimal Stock Screens. *Proceedings of the 3rd International Conference on Artificial Intelligence Applications on Wall Street.* 269-273.

Shin, K., & Han, I. (2004). A Genetic Algorithm Application in Bankruptcy Prediction Modeling. *Knowledge-Based Intelligent Information and Engineering Systems: 8th International Conference*, KES 2004, Wellington, New Zealand, September 20-25, 2004, Proceedings, Part II Editors: Mircea Gh. Negoita, Robert J. Howlett, Lakhmi C. Jain. Lecture Notes in Computer Science, 3214/2004.