Diagnosing Corporate Stability using Grammatical Evolution

Anthony Brabazon[1] and Michael O'Neill[2]

[1] Dept. Of Accountancy, University College Dublin, Ireland.
`Anthony.Brabazon@ucd.ie`
[2] Dept. Of Computer Science And Information Systems,
University of Limerick, Ireland.
`Michael.ONeill@ul.ie`

Abstract. Grammatical Evolution (GE) is a novel data-driven, model-induction tool, inspired by the biological gene-to-protein mapping process. This study provides an introduction to GE, and demonstrates the methodology by applying it to construct a series of models for the prediction of bankruptcy, employing information drawn from financial statements. Unlike prior studies in this domain, the raw financial information is not preprocessed into pre-determined financial ratios. Instead, the ratios to be incorporated into the classification rule are evolved from the raw financial data. This allows the creation and subsequent evolution of alternative ratio-based representations of the financial data. A sample of 178 publicly quoted, US firms, drawn from the period 1991 to 2000 are used to train and test the model. The best evolved model correctly classified 86 (77)% of the firms in the in-sample training set (out-of-sample validation set), one year prior to failure.

Keywords: Grammatical evolution, corporate failure prediction.

1 Introduction

The last decade has seen significant advances in the field of computational intelligence, leading to the development of powerful new modelling technologies. Generally, these technologies fall into three categories, those which are inspired by the workings of biological neurons (Neural Networks), those which are inspired by an evolutionary metaphor (Genetic Algorithms, Genetic Programming and Grammatical Evolution), and those which are inspired by studies of social interactions (Particle Swarm and Ant Colony models). While neural networks, and to a lesser extent genetic algorithms and ant-algorithms, have attracted considerable interest, other forms of computational intelligence have received relatively less attention.

Grammatical Evolution (GE) (O'Neill and Ryan, 2003), represents an evolutionary automatic programming methodology, and can be used to evolve 'rule sets'. These rule sets can be as general as a functional expression which produces a good mapping between a series of known input-output data vectors. A particular strength of the methodology is that the form of the model need not be specified *a priori* by the modeler. This is of particular utility in cases where the modeler has a theoretical or intuitive idea of the nature of the explanatory variables, but a weak understanding of the functional relationship between the explanatory and the dependent variable(s). GE does not require that the model form is linear, nor does the method require that the measure of model error used in model construction is a continuous or differentiable function. A key element of the methodology is the concept of a *Grammar*, which governs the creation of the rule sets. This paper introduces the GE methodology, and applies the methodology to construct a series of models for the prediction of bankruptcy, employing information drawn from financial statements.

Classification is a commonly encountered decision scenario in business. Examples include decisions as to whether or not to invest in a firm, whether to extend trade credit to a new customer, or whether to extend a bank loan. In each of these scenarios, the possibility of financial loss exists if a firm is incorrectly classified as being financially healthy, when in fact it is not. Corporate bankruptcy can impose significant private costs on many parties including shareholders, providers of debt finance, employees, suppliers, customers and auditors. Nonetheless, it must also be recognized that corporate failure is a natural component of a market economy, facilitating the recycling of financial, human and physical resources into more productive organizations (Easterbrook, 1990; Schumpeter, 1934). At an atomic level of analysis, many individuals including shareholders, providers of debt finance, employees, suppliers, customers, managers and auditors have an interest in the financial health of organizations as corporate failure can impose significant private costs on all these groups. It has been suggested that indicators of corporate failure can be present up to ten years prior to final failure (Hambrick and D'Aveni, 1988), providing an opportunity for construction of models which predict corporate failure.

Corporate failure can arise for many reasons. It may result from a single catastrophic event, or it may be the terminal point of a process of decline. Under the second perspective, corporate failure is a process which is rooted in management defects, resulting in poor decisions, leading to financial deterioration and finally corporate collapse (Altman, 1993; Hambrick and D'Aveni, 1988). Most attempts to predict corporate failure implicitly assume that management decisions critically impact on firm performance (Argenti, 1976). Although management decisions are not directly observable by external parties, their consequent affect on the financial health of the firm can be observed through their impact on the firm's financial ratios. Previous studies have utilized a wide variety of explanatory variables in the construction of corporate distress models, including data drawn from the financial statements of firms, data from financial markets, general macro-economic indicators, and non-financial, firm-specific information. In this study, we limit our attention to information drawn from financial statements.

1.1 Motivation for study

There are a number of reasons to suppose *a priori* that the use of an evolutionary automatic programming (EAP) approach such as GE, can prove fruitful in the financial prediction domain. The field is characterized by the lack of a strong theoretical framework and has a multitude of plausible, potentially interacting, explanatory variables. The first problem facing the modeler is the selection of a 'good' subset of these variables, and the second problem is the selection of an appropriate model form, representing a high-dimensional combinatorial problem. Evolutionary methodologies such as GE which can automate this process will be valuable.

To date, most attempts at developing models for the prediction of corporate failure have utilized a limited set of financial ratios (Altman, 1993). These ratios are generally selected on an *ad-hoc* basis by the modeler (Morris, 1997). Unfortunately, the number of ratios which can be calculated from a set of financial statements is large. A set of financial statements could contain several hundred numbers between the primary financial statements and the detailed notes accompanying the primary statements, resulting in a multitude of possible financial ratios which could potentially be included in a classification model. Most studies in this domain utilize similar financial ratios, typically justifying the choice of ratios by reference to earlier studies. This methodological approach leaves open the possibility that alternative, better, representations of the financial data exist. This study applies GE to this task, and novelly allows the modelling process to evolve different ratio representations from raw financial information.

The rest of this contribution is organized as follows. The next section provides an overview of the literature on corporate failure, followed by a section which describes Grammatical Evolution. We then outline the data set and methodology utilized. The following sections provide the results of the study followed by a number of conclusions.

2 Background

Research into the prediction of corporate failure using financial data, has a long history (Fitzpatrick, 1932; Smith and Winakor, 1935; Horrigan, 1965). Early statistical studies such as Beaver (1966), adopted a univariate methodology, identifying which accounting ratios had greatest classification accuracy in separating failing and non-failing firms. Although this approach did demonstrate classification power, it suffers from the shortcoming that a single weak financial ratio may be offset (or exacerbated) by the strength (or weakness) of other financial ratios. This issue was addressed in Altman (1968) by developing a multivariate LDA model and this was found to improve classification accuracy. Altman's (1968) discriminant function had the following form:

$$Z = .012X_1 + .014X_2 + .033X_3 + .006X_4 + .999X_5 \qquad (1)$$

where:
X_1 = working capital to total assets
X_2 = retained earnings to total assets
X_3 = earnings before interest and taxes to total assets
X_4 = market value of equity to book value of total debt
X_5 = sales to total assets

A later study by Altman, Haldeman and Narayanan (1977), using a larger dataset, selected the following set of explanatory variables (the study did not disclose the coefficients):

X_1 = return on assets (EBIT / Total Assets)
X_2 = stability of earnings
X_3 = debt service (EBIT / Total Interest)
X_4 = cumulative profitability (Retained Earnings / Total Assets)
X_5 = liquidity (Current Assets / Current Liabilities)
X_6 = capitalization (Equity / Total Capital)
X_7 = firm size (Total Assets)

Since the pioneering work of Beaver (1966) and Altman (1968), a vast array of methodologies have been applied for the purposes of corporate failure prediction. In the 1970s and 1980s, attention was focussed on Logit and Probit regression models (Gentry, Newbold and Whitford, 1985; Zmijewski, 1984; Ohlson, 1980). In more recent times, as the field of biologically-inspired computing has flourished, the methodologies applied to the domain of corporate failure prediction have expanded to include artificial neural networks (Shah and Murtaza, 2000; Serrano-Cinca, 1996; Wilson, Chong, Peel, 1995; Tam, 1991), genetic algorithms (Varretto, 1998; Kumar, Krovi and Rajagopalan, 1997), and grammatical evolution (Brabazon, O'Neill, Matthews and Ryan, 2002). Other methodologies applied to this problem include support vector machines (Fan and Palaniswami, 2000), rough sets (Zopounidis, Slowinski, Doumpos, Dimitras and Susmaga, 1999), and multicriteria decision analysis models (Zopounidis and Dimitras, 1998). Review studies covering much of the above literature can be found in Dimitras, Zanakis and Zopounidis (1996), and Morris (1997). Zhang, Hu, Patuwo and Indro (1999) provide a good review of prior applications of artificial neural networks to the domain of corporate failure.

2.1 Definition of Corporate Failure

No unique definition of corporate failure exists (Altman, 1993). Possible definitions range from failure to earn an economic rate of return on invested capital, to legal bankruptcy, followed by liquidation of the firm's assets. Typically, financial failure occurs when a firm incurs liabilities which cannot be repaid from liquid financial resources. However, this may represent the end of a period of financial decline, characterized by a series of losses and reducing liquidity. Any attempt to uniquely define corporate failure is likely to prove problematic. While few publicly quoted companies fail in any given year (Morris (1997) suggests that the rate is below 2% in the UK, and Zmijewski (1984) reports that this rate is less than 0.75% in the US), poorer performers are liable to acquisition by more successful firms. Thus, two firms may show a similar financial trajectory towards failure, but one firm may be acquired and 'turned-around' whilst the other may fail.

The definition of corporate failure adopted in this study is the court filing of a firm under Chapter 7 or Chapter 11 of the US Bankruptcy code. The selection of this definition provides an objective benchmark, as the occurrence (and timing) of either of these events can be determined through examination of regulatory filings. Chapter 7 of the US Bankruptcy code covers corporate liquidations and Chapter 11 covers corporate reorganizations, which usually follow a period of financial distress. Under Chapter 11, management is required to file a reorganization plan in bankruptcy court and seek approval for this plan. On filing the bankruptcy petition, the firm becomes a *debtor in possession.* Management continues to run the day-to-day business operations, but a bankruptcy court must approve all significant business decisions. In most cases, Chapter 11 reorganizations involve significant financial losses for both the shareholders (Russel, Branch and Torbey, 1999) and the creditors (Ferris, Jayaraman and Makhija, 1996) of the distressed firm. Moulton and Thomas (1993), in a study of the outcomes of Chapter 11 filings, found that there were relatively few successful reorganizations, despite a perception that some management teams were using Chapter 11 filings as a deliberate strategy for dealing with certain firm specific events such as onerous labor contracts or product liability claims. Out of a sample of 73 firms entering Chapter 11 between 1980 and 1986 that were examined in the study, only 44 were successfully reorganized and only 15 of these firms emerged from Chapter 11 with more than 50% of their pre-bankruptcy assets.

2.2 Explanatory variables utilized in prior literature

A comprehensive survey of the financial ratios employed in 47 journal articles on corporate failure is provided by Dimitras, Zanakis and Zopounidis (1996). If attention is restricted to ratios drawn from the financial statements of companies, five groupings are usually given prominence in the literature namely, liquidity, debt, profitability, activity, and size (Altman, 2000). Liquidity refers to the availability of cash resources to meet short-term cash requirements. Debt measures focus on the relative mix of funding provided by shareholders and lenders. Profitability considers the rate of return generated by a firm in relation to its size, as measured by sales revenue and/or asset base. Activity measures consider the operational efficiency of the firm in collecting cash, managing stocks and controlling its production or service process. Firm size provides information on both the sales revenue and asset scale of the firm and acts as a proxy metric on firm history (Levinthal, 1991). A range of individual financial ratios can represent the groupings of potential explanatory variables, each with slightly differing information content. The groupings are interconnected, as weak (or strong) financial performance in one area will impact on another. For example, a firm with a high level of debt may have lower profitability due to high interest costs. Whatever modelling methodology is applied in order to predict corporate distress, the initial problem is to select a quality set of model inputs, to preprocess these into a suitable ratio format, and then to combine the ratios using suitable weightings in order to construct a high quality classifier.

3 Grammatical Evolution

Evolutionary algorithms (EAs) operate on principles of evolution, usually being coarsely modelled on the theories of survival of the fittest and natural selection. In general, evolutionary algorithms can be characterized as:

$$x[t+1] = v(s(x[t])) \tag{2}$$

where $x[t]$ is the population of solutions at iteration t , $v(.)$ is the random variation operator (crossover and mutation), and $s(.)$ is the selection operator. Therefore the algorithm turns one population of candidate solutions into another, using selection, crossover and mutation. Selection exploits information in the current population, concentrating interest on 'high-fitness' solutions. Crossover and mutation perturb these solutions in an attempt to uncover better solutions, and these operators can be considered as general heuristics for exploration.

GE is a grammatical approach to Genetic Programming (GP) that can evolve computer programs (or 'rulesets') in any language, and a full description of GE can be found in (O'Neill and Ryan, 2003; O'Neill and Ryan, 2001; O'Neill, 2001; Ryan, Collins and O'Neill, 1998). Rather than representing the programs as syntax trees, as in Koza's GP (Koza, 1992), a linear genome representation is used. Each individual, a variable length binary string, contains in its codons (groups of 8 bits) the information to select production rules from a Backus Naur Form (BNF) grammar. In other words, an individual's binary string contains the instructions that direct a developmental process resulting in the creation of a program or 'rule'. As such, GE adopts a biologically-inspired, genotype-phenotype mapping process.

At present, the search element of the system is carried out by an evolutionary algorithm, although other search strategies with the ability to operate over binary or integer strings have also been used (O'Neill and Brabazon, 2004; O'Sullivan and Ryan, 2002). In particular, future advances in the field of evolutionary algorithms can be easily incorporated into this system.

3.1 The Biological Approach

The GE system is inspired by the biological process of generating a protein from the genetic material of an organism. Proteins are fundamental in the proper development and operation of living organisms and are responsible for traits such as eye color and height (Lewin, 2000).

The genetic material (usually DNA) contains the information required to produce specific proteins at different points along the molecule. For simplicity, consider DNA to be a string of building blocks called nucleotides, of which there are four, named A, T, G, and C, for adenine, tyrosine, guanine, and cytosine respectively. Groups of three nucleotides, called codons, are used to specify the building blocks of proteins. These protein building blocks are known as amino acids, and the sequence of these amino acids in a protein is determined by the sequence of codons on the DNA strand. The sequence of amino acids is very important as it determines the final three-dimensional structure of the protein, which in turn has a role to play in determining its functional properties.

Fig. 1. A comparison between the grammatical evolution system and a biological genetic system. The binary string of GE is analogous to the double helix of DNA, each guiding the formation of the phenotype. In the case of GE, this occurs via the application of production rules to generate the terminals of the compilable program. In the biological case by directing the formation of the phenotypic protein by determining the order and type of protein subcomponents (amino acids) that are joined together.

In order to generate a protein from the sequence of nucleotides in the DNA, the nucleotide sequence is first transcribed into a slightly different format, that being a sequence of elements on a molecule known as mRNA. Codons within the mRNA molecule are then translated to determine the sequence of amino acids that are contained within the protein molecule. The application of production rules to the non-terminals of the incomplete code being mapped in GE is analogous to the role amino acids play when being combined together to transform the growing protein molecule into its final functional three-dimensional form.

The result of the expression of the genetic material as proteins in conjunction with environmental factors is the phenotype. In GE, the phenotype is a sentence or sentences in the language defined by the input grammar. These sentences can take the form, for example, of functions, programs, or as in the case of this study, rule sets. The phenotype is generated from the genetic material (the genotype) by a process termed a genotype-phenotype mapping. This is unlike the standard method of generating a solution directly from an individual in an evolutionary algorithm by explicitly encoding the solution within the genetic material. Instead, a many-to-one mapping process is employed within which the robustness of the GE system lies. Figure 1 compares the mapping process employed in both GE and biological organisms.

3.2 The Mapping Process

When tackling a problem with GE, a suitable BNF (Backus Naur Form) grammar definition must first be defined. The BNF can be either the specification of an entire language or, perhaps more usefully, a subset of a language geared towards the problem at hand.

In GE, a BNF definition is used to describe the output language to be produced by the system. BNF is a notation for expressing the grammar of a language in the form of production rules. BNF grammars consist of terminals, which are items that can appear in the language, e.g. binary operators **+**, **-**, unary operators **Sin**, constants **1.0** etc. and non-terminals, which can be expanded into one or more terminals and non-terminals. For example from the grammar detailed below, <expr> can be transformed into one of four rules, i.e it becomes <expr><op><expr>, (<expr><op><expr>) (which is the same as the first, but surrounded by brackets), <pre-op>(<expr>), or <var>. A grammar can be represented by the tuple $\{N, T, P, S\}$, where N is the set of non-terminals, T the set of terminals, P a set of production rules that maps the elements of N to T, and S is a start symbol which is a member of N. When there are a number of productions that can be applied to one element of N the choice is delimited with the '|' symbol. For example,

```
N = { <expr>, <op>, <pre_op> }
T = {Sin, +, -, /, *, X, 1.0, (, )}
S = <expr>
```

And P can be represented as:

```
(A) <expr> ::= <expr> <op> <expr>      (0)
            | ( <expr> <op> <expr> ) (1)
            | <pre-op> ( <expr> )    (2)
            | <var>                  (3)

(B) <op> ::= +       (0)
          | -       (1)
          | /       (2)
          | *       (3)

(C) <pre-op> ::= Sin

(D) <var> ::= X      (0)
           | 1.0    (1)
```

The program, or sentence(s), produced will consist of elements of the terminal set T. The grammar is used in a developmental approach whereby the evolutionary process evolves the production rules to be applied at each stage of a mapping process, starting from the start symbol, until a complete program is formed. A complete program is one that is comprised solely from elements of T.

As the BNF definition is a plug-in component of the system, it means that GE can produce code in any language thereby giving the system a unique flexibility.

For the above BNF, Table 1 summarizes the production rules and the number of choices associated with each.

Rule no.	Choices
A	4
B	4
C	1
D	2

Table 1. The number of choices available from each production rule.

The genotype is used to map the start symbol onto terminals by reading codons of 8 bits to generate a corresponding integer value, from which an appropriate production rule is selected by using the following mapping function:

$$Rule = Codon\ Value\ \%\ No.\ Rule\ Choices \tag{3}$$

Consider the following rule from the given grammar i.e., given the non-terminal *op*, which describes the set of binary operators that can be used, there are four production rules to select from.

```
(B) <op> :: =    +              (0)
               | -              (1)
               | /              (2)
               | *              (3)
```

If we assume the codon being read produces the integer 6, then

$$6\ \%\ 4\ =\ 2$$

would select rule (2) /. Each time a production rule has to be selected to transform a non-terminal, another codon is read. In this way the system traverses the genome.

During the genotype-to-phenotype mapping process, it is possible for individuals to run out of codons, and in this case we wrap the individual and reuse the codons. This is quite an unusual approach in EAs, as it is entirely possible for certain codons to be used two or more times. This technique of wrapping the individual draws inspiration from the gene-overlapping phenomenon that has been observed in many organisms (Lewin, 2000).

In GE, each time the same codon is expressed it will always generate the same integer value, but, depending on the current non-terminal to which it is being applied, it may result in the selection of a different production rule. This feature is referred to as intrinsic polymorphism. Crucially, however, each time a particular individual is mapped from its genotype to its phenotype, the same output is generated. This is the case because the same choices are made each time. However, it is possible that an incomplete mapping could occur, even after several wrapping events, and in this case the individual in question is given the lowest possible fitness value. The selection and replacement mechanisms then operate accordingly to increase the likelihood that this individual is removed from the population.

An incomplete mapping could arise if the integer values expressed by the genotype were applying the same production rules repeatedly. For example, consider an individual with three codons, all of which specify rule 0 from below,

```
(A) <expr> :: =  <expr><op><expr>        (0)
                |(<expr><op><expr>)      (1)
                |<pre-op>(<expr>)        (2)
                |<var>                   (3)
```

even after wrapping the mapping process would be incomplete and would carry on indefinitely unless stopped. This occurs because the nonterminal `<expr>` is being mapped recursively by production rule 0, i.e., it becomes `<expr><op><expr>`. Therefore, the leftmost `<expr>` after each application of a production would itself be mapped to a
`<expr><op><expr>`, resulting in an expression continually growing as follows: `<expr><op><expr><op><expr><op><expr>` etc.

Such an individual is dubbed invalid as it will never undergo a complete mapping to a set of terminals. For this reason we impose an upper limit on the number of wrapping events that can occur. It is clearly essential that stop sequences are found during the evolutionary search in order to complete the mapping process to a functional program. The stop sequence being a set of codons that result in the non-terminals being transformed into elements of the grammars terminal set.

Beginning from the left hand side of the genome then, codon integer values are generated and used to select rules from the BNF grammar, until one of the following situations arise:

1. A complete program is generated. This occurs when all the non-terminals in the expression being mapped are transformed into elements from the terminal set of the BNF grammar.
2. The end of the genome is reached, in which case the *wrapping* operator is invoked. This results in the return of the genome reading frame to the left hand side of the genome once again. The reading of codons will then continue, unless an upper threshold representing the maximum number of wrapping events has occurred during this individual's mapping process.
3. In the event that a threshold on the number of wrapping events has occurred and the individual is still incompletely mapped, the mapping process is halted, and the individual is assigned the lowest possible fitness value.

To reduce the number of invalid individuals being passed from generation to generation, a steady state replacement mechanism is employed. One consequence of the use of a steady state method is its tendency to maintain fit individuals at the expense of less fit, and in particular, invalid individuals.

In this study, the GE algorithm uses a steady state replacement mechanism, such that, two parents produce two children the best of which replaces the worst individual in the current population, if the child has greater fitness. The standard genetic operators of bit mutation (probability of 0.01), and crossover (probability of 0.9) are adopted. A series of functions, are pre-defined as are a series of mathematical operators. A population of initial rule-sets (programs) are randomly generated, and by means of an evolutionary process, these are improved. No explicit model specification is assumed *ex-ante*, although the choice of mathematical operators defined in the grammar do place implicit limitations on the model specifications amongst which GE can search.

4 Problem Domain & Experimental Approach

This section describes both the data utilized by, and the model development process adopted in, this study.

4.1 Sample Definition and Model Data

A total of 178 firms were selected judgemental (89 failed, 89 non-failed), from the Compustat Database. Firms from the financial sector were excluded on grounds of lack of comparability of their financial ratios with other firms in the sample. The criteria for selection of the failed firms were:

i. Inclusion in the Compustat database in the period 1991-2000
ii. Existence of required data for a period of three years prior to entry into Chapter 7 or Chapter 11
iii. Sales revenues must exceed $1M

The first criterion limits the study to publicly quoted, US corporations. For every failing firm, a matched non-failing firm is selected. Failed and non-failed firms are matched both by industry sector and size (sales revenue three years prior to failure). It is recognized that the use of an equalized, matched sample entails sampling bias and eliminates firm size and industry nature as potential explanatory variables (see Morris (1997) for a detailed discussion of these points), and it is noted that utilizing an unmatched sample imposes its own bias. The set of 178 matched firms are randomly divided into model building (128 firms) and out-of-sample (50 firms) data-sets, with each data-set consisting of matched pairs of failed and non-failed firms. The dependant variable is binary (0,1), representing either a non-failed or a failed firm. In this study, rather than pre-specifying financial ratios, GE can create ratios from raw financial data. We have initially restricted our choice of raw financial data to the following twelve items, extracted from their annual financial statements:

i. Sales
ii. Net Income
iii. Gross Profit
iv. EBIT
v. EBITDA
vi. Total Assets
vii. Total Current Assets

viii. Total Liabilities
ix. Total Current Liabilities
x. Total Long-term Debt
xi. Cash from Operations
xii. Free-Cash Flow

This information was collected for each firm for the three years prior to entry, either by it or its matched firm, into Chapter 7 or Chapter 11 (denoted as T-3, T-2 and T-1, where T-3 is three years prior to failure). The date of entry into Chapter 7 or Chapter 11 was determined by examining US Securities and Exchange Commission (SEC) regulatory filings for each firm.

Three Grammars are employed, in order to examine the impact on predictive accuracy of allowing GE to evolve classification rules of varying complexity. The three Grammars are as follows:

```
Grammar 1

<lc> ::= output = <coeff> *  ( ( <var> ) / ( <var> ) );
<coeff> ::=  ( <coeff> ) <op> ( <coeff> )
          | <float>
<var> ::= Sales  | Net Income  | Gross Profit
        | EBIT  | EBITDA  | Total Assets
        | Total Current Assets  | Total Liabilities
        | Total Current Liabilities | Total Long Term Debt
        | Cash From Operations | Free Cash Flow
<op> ::= +
       | -
<float> ::= 20 | -20 | 10 | -10 | 5 | -5 | 4 | -4
          | 3 | -3 | 2 | -2 | 1 | -1 | .1 | -.1

Grammar 2

<lc> ::= output = <expr> ;
<expr> ::= ( <expr> ) + ( <expr> )
         | <coeff> * ( <var> / <var> )
<var> ::= Sales  | Net Income  | Gross Profit
        | EBIT  | EBITDA  | Total Assets
        | Total Current Assets  | Total Liabilities
        | Total Current Liabilities | Total Long Term Debt
        | Cash From Operations | Free Cash Flow
<coeff> ::=  ( <coeff> ) <op> ( <coeff> )
          | <float>
<op> ::= +
       | -
<float> ::= 20 | -20 | 10 | -10 | 5 | -5 | 4 | -4
          | 3 | -3 | 2 | -2 | 1 | -1 | .1 | -.1

Grammar 3

<lc> ::= output = <expr> ;
<expr> ::= ( <expr> ) + ( <expr> )
         |  <coeff> * ( <ratio> / <var> )
<ratio> ::= <ratio> <op> <ratio>
          | <var>
<var> ::= Sales  | Net Income  | Gross Profit
        | EBIT  | EBITDA  | Total Assets
        | Total Current Assets  | Total Liabilities
        | Total Current Liabilities | Total Long Term Debt
        | Cash From Operations | Free Cash Flow
<coeff> ::=  ( <coeff> ) <op> ( <coeff> )
          | <float>
<op> ::= +
```

```
       | -
<float> ::= 20 | -20 | 10 | -10 | 5 | -5 | 4 | -4
          | 3 | -3 | 2 | -2 | 1 | -1 | .1 | -.1
```

Grammar one permits the construction of a predictive rule consisting of a single ratio, formed from any two discrete pieces of raw financial data. This ratio can be rescaled as required by an evolved coefficient parameter. In essence, this Grammar searches for the best univariate predictive model. Grammar two permits the construction of predictive rules which chain ratios together, producing linear rules of the form:

```
output = coefficient * Ratio X + coefficient * Ratio Y + ...
```

In each of these Grammars, only ratios of the form $\frac{a}{b}$, where a and b are discrete pieces of financial data are permitted. Grammar three allows the construction of a linear chain of ratios, where the ratios can take the form $\frac{a+b+\cdots}{x}$, greatly increasing the number of possible ratios that can be formed from the raw data. The output from the classifier is post-processed using a cut-off value of 0.50 to produce a classification.

4.2 Selection of Fitness Function

Most studies of corporate failure adopt classification accuracy as their error (fitness) criterion. If misclassification costs are symmetric, the sample error rate is:

$$error_{sample} = \frac{m_1 + m_2}{n_1 + n_2} \tag{4}$$

where m_1 is the number of failed firms (out of n_1) in the sample that are misclassified and m_2 is the number of non-failed firms (out of n_2) in the sample that are misclassified. It is recognized that misclassification costs will not always be symmetric, and in this case, overall classification accuracy will not be an adequate measure of model performance because it does not reflect the relative costs of misclassifications between the two groups. However, misclassification costs cannot be defined uniquely, as their relative sizes will vary depending on the identity of decision-maker. This study assumes that misclassification costs are symmetric, but alternative treatments could be easily incorporated in the fitness function.

5 Results

The results from our experiments are now provided. Each of the GE experiments is run with a population size of 500, for 100 generations, with one-point crossover at a probability of 0.9, one point bit mutation at a probability of 0.01, roulette selection, and steady-state replacement. 30 independent runs of the GE algorithm were undertaken in each experiment, and the reported results include the best evolved individual, the mean best fitness and mean average fitness over the 30 runs in each case.

Three series of models were constructed for each Grammar, using raw financial information drawn from one, two and three years (T-1, T-2 and T-3) prior to failure. In all cases the reported accuracies are determined across three recuts of the dataset into training and test (out-of-sample) data, maintaining an equal balance of failed/non-failed companies in the resulting training and testing datasets. The average of the best individuals evolved, across all three data recuts, for each period are reported in Table 2.[1]

Years Prior to Failure	**Grammar 1 (%)**	**2 (%)**	**3 (%)**
1	82.67 (70.67)	85.67 (73.33)	86.00 (76.67)
2	77.33 (68.67)	80.33 (73.33)	80.33 (73.33)
3	71.67 (57.33)	73.00 (62.67)	75.00 (56.67)

Table 2. The in-sample (out-of-sample) classification accuracies for the best individuals, averaged across the three recuts of the dataset, in each Grammar for the three years prior to failure.

As expected, the classification accuracies improve as the date of failure approaches, ranging from approximately 85% (in-sample) at T-1, to approximately 73% (in-sample) at T-3. Across the three Grammars, Grammar 1 which can only evolve a univariate ratio, is slightly out-performed both in and out-of-sample by Grammars 2 and 3. Neither Grammar 2 or 3 clearly dominate each other, suggesting that the ability of Grammar 3 to evolve complex ratio forms has not led to the generation of better quality classifiers.

To assess the overall hit-ratio of the developed models (out-of-sample), Press's Q statistic (Hair, Anderson, Tatham and Black, 1998) was calculated for each model. For T-1 and T-2, the null hypothesis, that the obtained out-of sample classification accuracies are not significantly better than those that could occur by chance alone, was rejected at the 5% level. A t-test of the hit-ratios also rejected a null hypothesis that the classification accuracies were no better than chance at the 5% level for both T-1 and T-2.

Additional metrics were collected on the positive accuracy (correct prediction of non-failure) and negative accuracy (correct prediction of failure) for each of the models. Table 3 provides these for the out-of-sample datasets. All reported results are averaged across the three different randomizations of the dataset for each year. In the case of the univariate Grammar (Grammar 1), the results for T-2 and T-3 display asymmetry, with the evolved classifier identifying surviving companies more easily than failing companies. Generally the results for Grammars 2 and 3 are reasonably symmetric, but in line with prior findings (Altman,

[1] The best individual is defined with reference to performance on the in-sample data.

1993) the evolved classifiers find it slightly easier to correctly classify non-failing than failing companies.

Years Prior	Grammar		
to Failure	**1 (%)**	**2 (%)**	**3 (%)**
1	69.33 (72.00)	77.33 (69.33)	78.67 (74.67)
2	90.67 (46.67)	73.33 (73.33)	74.67 (72.00)
3	65.33 (49.33)	65.33 (60.00)	65.33 (48.00)

Table 3. Positive (negative) classification accuracy out-of-sample for the best individuals, averaged across all three recuts.

Graphs of the evolution of fitness during the training run, for the best individual for each Grammar, time period and data cut are provided in Figures 2, 3 and 4. Graphs are also provided of the fitness evolution for the average of the best individuals across all 30 runs. The graphs suggest that the choice of 100 generations was sufficient to allow for evolution of quality classifiers for each Grammar, with most of the gains in evolutionary fitness occurring in the first 50 generations.

Fig. 2. Best (left) and average (right) fitness values for all three recuts, for one year prior to failure, for all three grammars on the in-sample dataset.

In order to provide insight into the form of the classifier rules evolved by the Grammars, Table 4 lists the best classifiers (defined as producing the 'best' classification performance on the training dataset) for each Grammar for T-1 to T-3. In some cases, there was more than one rule producing equivalent

Fig. 3. Best (left) and average (right) fitness values for all three recuts, for two years prior to failure, for all three grammars on the in-sample dataset.

classification accuracy. In these cases, the best classifier listed in the table was chosen judgementally. The interpretation of these classifier rules is considered in section 5.1.

Years Prior to Failure	Best Classifier
Grammar 1	
1	- 2*(Total Current Liabilities / Net Income)
2	1 -19*(Net Income / Total Current Liabilities)
3	1 - 7*(Net Income / Total Current Liabilities)
Grammar 2	
1	(-1 - 6*(Cash From Operations / Sales)) + 2*(Total Liabilities / Total Assets)
2	1*(Total Liabilities / Total Assets) -5*(EBIT / Sales)
3	-2*(Cash From Operations / Gross Profit) + 3 -29.9*(Net Income / Total Current Liabilities)
Grammar 3	
1	-4*((Total Assets - Total Current Assets + EBIT) / Sales) + -20*((EBITDA - Net Income) / Net Income)
2	(-5*(Net Income / Total Current Liabilities)) + .1*(Sales / Gross Profit)
3	(3 - 20*(Net Income / Total Current Liabilities)) -20*(Cash From Operations / Sales)

Table 4. The best classifiers evolved for each of the years and Grammars.

5.1 Discussion

Despite using financial data drawn from a wide variety of industrial sectors, the evolved models showed a capability to discriminate between failing and non-

Fig. 4. Best (left) and average (right) fitness values for all three recuts, for three years prior to failure, for all three grammars on the in-sample dataset.

failing firms, most notably in the two years prior to corporate collapse. The risk factors suggested by each classifier in Table 4 differ somewhat but present plausible findings.

In Grammar 1, the best evolved ratios for all three time periods contain an earnings (profit) component, and relate the size of the company's profits to its short-term liabilities. In all cases, the sign of the ratio coefficient is plausible, with lower (or negative) earnings indicating greater risk of failure.

The classifier rule for T-1 under Grammar 2 utilizes a combination of ratios which focus on the ability of the company to generate cash from its sales, and the size of the debt of the company relative to its asset base. The coefficients are plausible with strong cash generation and low indebtedness suggesting a financially healthy company. For T-2, the classifier concentrates on the ability of the company to generate profits from its sales, and the size of the debt of the company relative to its asset base. Strong profitability and low indebtedness indicate financial health. In the case of T-3, the evolved ratios concentrate on the cash generation ability of the company, and the level of short-term debt relative to its profitability. High cash generation relative to sales, and low levels of short-term debt relative to profits, indicate a healthy company.

The evolved classifiers under Grammar 3 are similar in form for T-3 and T-2, emphasizing the cash generation ability of the company and the level of profit relative to the company's short-term liabilities. In both cases, strong cash generation and low levels of short-term debt relative to the company's profit suggest a financially viable company. For T-1, the evolved classifier concentrates on the level of profit generated by the company relative to its short-term liabilities. Again, high levels of profit relative to short-term debt indicate a financially strong company.

The evolved rules and their related coefficients, across all Grammars and across all time periods, are in accordance with financial intuition, and do not display evidence of merely resulting from data-mining. The results also suggest that Grammars 2 and 3, which have the capability to evolve complex combinations of the financial data, did not greatly out-perform the simpler models produced by Grammar 1. Considering the individual Grammars, it interesting that despite the potential of Grammars 2 and 3 to generate long, complex ratio chains, this bloating did not occur and the evolved classifiers are reasonably concise in form. We also note that the evolved classifiers (unlike those created by means of a neural network methodology, for example) are amenable to human interpretation.

6 Conclusions

In this paper a novel methodology, GE, was introduced and applied for the purposes of prediction of corporate failure. It is noted that this novel methodology has general utility for rule-induction applications. GE was found to be able to evolve quality classifiers for corporate failure from raw financial information. In performing this task, GE was required to evolve its own ratio representation of the financial data, rather than being supplied with modeler-defined financial ratios as is typically the case in studies of corporate failure.

In assessing the performance of the developed models, a number of caveats must be borne in mind. The premise underlying this paper and all empirical work on corporate failure prediction, is that corporate failure is a process, commencing with poor management decisions, and that the trajectory of this process can be tracked using accounting ratios. This approach does have inherent limitations. It will not forecast corporate failure which results from a sudden environmental event. Although not undertaken in this study, the incorporation of non-financial qualitative explanatory variables or variables related to the firm's share price performance could further improve classification accuracy. Finally, the firms sampled in this study are relatively large and are publicly quoted. Thus, the findings of this study may not extend to small businesses. Despite these limitations, the high economic and social costs of corporate failure imply that models which can indicate declining financial health will have utility. Given the lack of a clear theory underlying corporate failure, empirical modelling usually adopts a combinatorial approach, a task for which GE is well suited.

References

1. Altman, E. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy, *Journal of Finance*, Vol. 23, No. 1, pp. 589-609.
2. Altman, E., Haldeman, R. and Narayanan, P. (1977). ZETA Analysis: A new model to identify bankruptcy risk of corporations, *Journal of Banking and Finance*, Vol. 1, No, 1, pp. 29-54.
3. Altman, E. (1993). *Corporate Financial Distress and Bankruptcy*, New York: John Wiley and Sons Inc.

4. Altman, E. (2000). Predicting Financial Distress of Companies: Revisiting the Z-score and Zeta models, *http://www.stern.nyu.edu/ ealtman/Zscores.pdf, October 2001.*
5. Argenti, J. (1976). *Corporate Collapse: The Causes and Symptoms*, London: McGraw-Hill.
6. Back, B., Laitinen, T., Sere, K. and van Wezel, M. (1996). Chosing Bankruptcy Predictors using Discriminant Analysis, Logit Analysis and Genetic Algorithms, *Technical Report no. 40, Turku Centre for Computer Science*, Turku School of Economics and Business Administration.
7. Beaver, W. (1966). Financial Ratios as Predictors of Failure, *Journal of Accounting Research - Supplement: Empirical Research in Accounting*, pp. 71-102.
8. Brabazon, T., O'Neill, M., Matthews, R., and Ryan, C. (2002). Grammatical Evolution and Corporate Failure Prediction, In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, Spector et. al. Eds., July 9-13, 2002, pp. 1011-1019, New York: Morgan Kaufmann.
9. Dimitras, A., Zanakis, S. and Zopounidis, C. (1996). A survey of business failures with an emphasis on prediction methods and industrial applications, *European Journal of Operational Research*, Vol. 90, No. 3, pp. 487-513.
10. Easterbrook, F. (1990). Is Corporate Bankruptcy Efficient?, *Journal of Financial Economics*, Vol. 27, pp. 411-417.
11. Fan, A. and Palaniswami, M. (2000). A new approach to corporate loan default prediction from financial statements, in *Proceedings of Computational Finance / Forecasting Financial Markets Conference (CF/FFM-2000)*, May 2000, London.
12. Ferris, S., Jayaraman, N. and Makhija, A. (1996). The Impact of Chapter 11 filings on the Risk and Return of Security Holders, 1979-1989, *Advances in Financial Economics*, Vol. 2, pp. 93-118.
13. Fitzpatrick, P. (1932). *A Comparison of the Ratios of Successful Industrial Enterprises with Those of Failed Companies*, Washington: The Accountants' Publishing Company.
14. Gentry, J., Newbold, P. and Whitford, D. (1985). Classifying Bankrupt Firms with Funds Flow Components, *Journal of Accounting Research*, Vol. 23, No. 1, pp. 146-160.
15. Hambrick, D. and D'Aveni, R. (1988). Large Corporate Failures As Downward Spirals, *Administrative Science Quarterly*, Vol. 33, pp. 1-23.
16. Hair, J., Anderson, R., Tatham, R. and Black, W. (1998). *Multivariate Data Analysis*, Upper Saddle River, New Jersey: Prentice Hall.
17. Horrigan, J. (1965). Some empirical bases of financial ratio analysis, *The Accounting Review*, July 1965, pp. 558-568.
18. Koza, J. (1992). *Genetic Programming*, Massachusetts: MIT Press.
19. Kumar, N., Krovi, R. and Rajagopalan, B. (1997). Financial decision support with hybrid genetic and neural based modelling tools, *European Journal of Operational Research*, Vol. 103, No. 2, pp. 339-349.
20. Levinthal, D. (1991). Random Walks and Organisational Mortality, *Administrative Science Quarterly*, Vol. 36, No. 3, pp. 397-420.
21. Lewin, Benjamin. (2000). *Genes VII*. Oxford University Press.
22. Morris, R. (1997). *Early Warning Indicators of Corporate Failure: A critical review of previous research and further empirical evidence*, London: Ashgate Publishing Limited.
23. Moulton, W. and Thomas, H. (1993). Bankruptcy As a Deliberate Strategy: Theoretical Considerations and Empirical Evidence, *Strategic Management Journal*, Vol. 14, pp. 125-135.

24. Ohlson, J. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy, *Journal of Accounting Research*, Vol. 18, No. 1, pp. 109-131.
25. O'Neill, M., Brabazon, A. (2004). Grammatical Swarm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2004*. To appear.
26. O'Neill M., Ryan C. (2003). *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*, Boston: Kluwer Academic Publishers.
27. O'Neill M. (2001). Automatic Programming in an Arbitrary Language: Evolving Programs with Grammatical Evolution, PhD thesis, University of Limerick, Ireland, 2001.
28. O'Neill M., Ryan C. (2001).Grammatical Evolution, *IEEE Trans. Evolutionary Computation*, Vol. 5, No. 4, pp. 349-358.
29. O'Sullivan J., Ryan C. (2002). An investigation into the use of different search strategies with Grammatical Evolution, *Proceedings of EuroGP 2002, in Lecture Notes in Computer Science (2278): Genetic Programming*, edited by J. Foster, E. Lutton, J. Miller, C. Ryan and A. Tettamanzi, Springer-Verlag, Berlin, pp. 103-113.
30. Russel, P., Branch, B. and Torbey, V. (1999). Market Valuation of Bankrupt Firms: is there an anomaly?, *Quarterly Journal of Business and Economics*, Vol. 38, pp. 55-76.
31. Ryan C., Collins J.J., O'Neill M. (1998). Grammatical Evolution: Evolving Programs for an Arbitrary Language. *Lecture Notes in Computer Science 1391, Proceedings of the First European Workshop on Genetic Programming*, pp. 83-95, Berlin: Springer-Verlag.
32. Schumpeter, J. (1934). *The Theory of Economic Development*, Cambridge, MA: Harvard Business Press.
33. Serrano-Cinca, C. (1996). Self organizing neural networks for financial diagnosis, *Decision Support Systems*, Vol. 17, No. 3, pp. 227-238.
34. Shah, J. and Murtaza, M. (2000). A Neural Network Based Clustering Procedure for Bankruptcy Prediction, *American Business Review*, Vol. 18, No. 2, pp. 80-86.
35. Smith, R. and Winakor, A. (1935). Changes in the Financial Structure of Unsuccessful Corporations, *University of Illinois, Bureau of Business Research, Bulletin No. 51.*
36. Tam, K. (1991). Neural network models and the prediction of bank bankruptcy, *Omega*, Vol. 19, No. 5, pp. 429-445.
37. Varetto, F. (1998). Genetic algorithms in the analysis of insolvency risk, *Journal of Banking and Finance*, Vol. 22, No. 10, pp. 1421-1439.
38. Wilson, N., Chong, K. and Peel, M. (1995). Neural Network Simulation and the Prediction of Corporate Outcomes: Some Empirical Findings, *International Journal of the Economics of Business*, Vol. 2, No. 1, pp. 31-50.
39. Zhang, G., Hu, M., Patuwo, B. and Indro, D. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis, *European Journal of Operational Research*, Vol. 116, pp. 16-32.
40. Zmijewski, M. (1984). Methodological Issues Related to the Estimation of Financial Distress Prediction Models, *Journal of Accounting Research-Supplement*, pp. 59-82.
41. Zopounidis, C. and Dimitras, A. (1998). *Multicriteria Decision Aid Methods for the Prediction of Business Failure*, Dordrecht: Kluwer Academic Publishers.
42. Zopounidis, C., Slowinski, R, Doumpos, M., Dimitras, A. and Susmaga, R. (1999). Business failure prediction using rough sets: A comparision with multivariate analysis techniques, *Fuzzy Economic Review*, Vol. 4, No. 1, pp. 3-33.