Credit Classification Using Grammatical Evolution

Anthony Brabazon and Michael O'Neill

University College Dublin, Ireland.
Anthony.Brabazon@ucd.ie

University of Limerick, Ireland.
Michael.ONeill@ul.ie

Keywords: Grammatical evolution, credit rating, bond rating

Received: May 25, 2004

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 model the corporate bond-issuer credit rating process, using information drawn from the financial statements of bond-issuing firms. Financial data and the associated Standard & Poor's issuer-credit ratings of 791 US firms, drawn from the year 1999/2000 are used to train and test the model. The best developed model was found to be able to discriminate in-sample (out-of-sample) between investment-grade and junk bond ratings with an average accuracy of 87.59 (84.92)% across a five-fold cross validation.

Povzetek:

1 Introduction

Grammatical Evolution (GE) [1], 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. Neither is GE a black box method. As such the evolved rules (taking the form of symbolic expressions in this instance) are amenable to human interpretation and consequently have the potential to enhance our understanding of the problem domain.

A key element of the methodology is the concept of a *Grammar*, which governs the creation of the rule sets. This paper describes the GE methodology, and applies the methodology to accurately model the corporate bond rating process.

Most large firms employ both share and debt capital to provide long-term finance for their operations. The debt capital may be provided by a bank, or may be obtained by selling bonds directly to investors. As an example of the scale of US bond markets, the value of bonds issued in the first quarter of 2003 totalled $1.70 trillion [2]. A bond can be defined as a 'debt security which constitutes a promise by the issuing firm, to pay a stated rate of interest based on the face value of the bond, and to redeem the bond at this face value at maturity.' When a publicly-traded company wants to issue traded debt (bonds), it must obtain a credit rating for the issue from at least one recognised rating agency (Standard and Poor's (S&P), Moody's or Fitches'). The credit rating represents the rating agency's opinion, at a specific date, of the creditworthiness of a borrower in general (an issuer credit rating), or in respect of a specific debt issue (a bond credit rat-

ing). Therefore it serves as a surrogate measure of the risk of non-payment of interest or capital of a bond. These ratings impact on the borrowing cost and the marketability, of issued bonds.

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 this domain.

In common with the related corporate failure prediction problem [3], a feature of the bond-rating problem is that there is no clear theoretical framework for guiding the choice of explanatory variables, or model form. Rating agencies assert that their credit rating process involves consideration of both financial and non-financial information about the firm and its industry, but the precise factors utilised, and the related weighting of these factors, are not publicly disclosed by the rating agencies. In the absence of an underlying theory, most published work on credit rating prediction employs a data-inductive modelling approach, using firm-specific financial data as explanatory variables, in an attempt to recover the model used by the rating agencies. This produces a high-dimensional combinatorial problem, as the modeller is attempting to uncover a good set of model inputs, and model form, giving rise to particular potential for an evolutionary automatic programming methodology such as GE. In this initial application of GE to modelling credit rating, we restrict attention to the binary classification case (discriminating between investment grade vs junk grade ratings). This will be extended to the multi-class case in future work. It is noted that a limited number of studies have applied a grammar-based methodology to constrain the search space for classification rules [3, 4, 5, 6]. This study extends this methodology into the domain of bond-rating.

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

2 Bond Rating

Several categories of individuals would be interested in a model that could produce accurate estimates of bond ratings. Such a model would be of interest to firms that are considering issuing debt as it would enable them to estimate the likely return investors would require if the debt was issued, thereby providing information for the pricing of the bonds. The model could also be used to assess the credit-worthiness of firms that have not issued debt and hence do not already have a published bond rating. This information would be useful to bankers or other companies that are considering whether they should extend credit to that firm. Much rated debt is publicly traded on stock markets, and bond ratings are typically changed infrequently. An accurate bond-rating prediction model could indicate whether the current rating of a bond is still justified. To the extent that an individual investor could predict a bond rerating before other investors foresee it, this may provide a trading edge. In addition, the recent introduction of credit-risk derivatives allows investors to buy protection against the risk of the downgrade of a bond [7]. The pricing of such derivative products requires a quality model for estimating the likelihood of a credit rating change.

2.1 Bond Rating Notation

Although the precise notation used by individual rating agencies to denote the creditworthiness of a bond or issuer varies, in each case the rating is primarily denoted by a discrete, mutually exclusive, 'letter grade'. Taking the rating structure of S&P as an example, the ratings are broken down into 10 broad classes. The highest rating is denoted AAA, and the ratings then decrease in the following order, AA, A, BBB, BB, B, CCC, CC, C, D. Ratings between AAA and BBB (inclusive) are deemed to represent *investment grade*, with lower quality ratings deemed to represent debt issues with significant speculative characteristics (*junk bonds*). A 'C' grade represents a case where a bankruptcy petition has been filed, and a 'D' rating represents a case where the borrower is currently in default on their financial obligations. As would be expected, the probability of default depends strongly on the initial rating which a bond receives (see table 1).

Initial Rating	**Defaults (%)**
AAA	0.52
AA	1.31
A	2.32
BBB	6.64
BB	19.52
B	35.76
CCC	54.38

Table 1: Rate of default by initial rating category (1987-2002)(from [8]).

Ratings from AAA to CCC can be modified by the addition of a + or a -, to indicate at which end of the rating category the bond rating falls. An initial rating is prepared when a bond is being issued, and this rating is periodically reviewed thereafter by the rating agency. Bonds (or issuers) may be re-rated upwards (upgrade) or downwards (downgrade) if firm or environmental circumstances change. A re-rating of a bond below investment grade to junk bond status (such bonds are colorfully termed 'a fallen angel') may trigger a significant sell-off as many institutional investors are only allowed, by external or self-imposed regulation, to hold bonds of investment grade. The practical affect of a bond (or issuer) being assigned a lower rather than a higher rating is that its perceived riskiness in the eyes of potential investors increases, and consequently the required interest yield of the bond rises.

2.2 Prior Literature

In essence, the objective of constructing a model of bond ratings, is to produce a model of rating agency behaviour, using publicly available information. A large literature exists on bond-rating prediction. Earliest attempts utilised statistical methodologies such as linear regression (OLS) [9], multiple discriminant analysis [10], the multinomial logit model [11], and ordered-probit analysis [12]. The results from these studies varied, and typically results of about 50-60% prediction accuracy (out-of-sample) were obtained, using financial data as inputs. With the advent of artificial intelligence and machine learning, the range of techniques applied to predict bond ratings has expanded to include neural networks [13]. In the case of prior neural network research, the predictive accuracy of the developed models has varied. Several studies employed a binary predictive target and reported good classification accuracies. For example, [14] used a neural network to predict AA or non-AA bond ratings, and obtained an accuracy of approximately 83.3%. However, a small sample size (47 companies) was adopted in the study, making it difficult to generalise strongly from its results.

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 [15]. In general, evolutionary algorithms can be characterized as:

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

where $x[t]$ is the population of solutions at iteration t , $v(.)$ is the random variation operator (crossover and mutation), $s(.)$ is the selection for reproduction operator, and r is the replacement operator which determines which of the parents and children survive into the next generation. 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 [1, 16, 17, 18]. Rather than representing the programs as syntax trees, as in Koza's GP [19], 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 [1, 5]. The GE system possesses a modular structure (see figure 1) which will allow future advances in the field of evolutionary algorithms to be easily incorporated.

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 [20].

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.

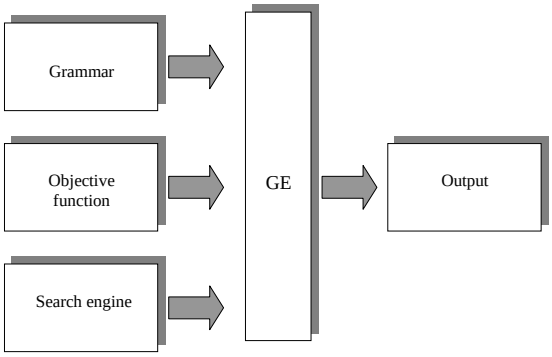

Figure 1: Modular structure of grammatical evolution

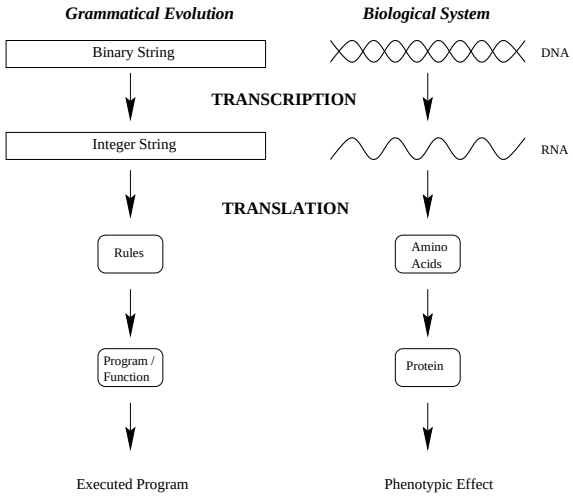

Figure 2: A comparison between the grammatical evolution system and a biological genetic system. The binary string of GE is analogous to the double helix of DNA, each guiding the formation of the phenotype. In the case of GE, this occurs via the application of production rules to generate the terminals 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 2 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 2 summarizes the production rules and the number of choices associated with each.

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

Table 2: 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:

$$Rule = Codon\ Value\ \%\ No.\ Rule\ Choices \quad (2)$$

where % is the MOD function which returns the remainder after a division operation (e.g. 4 % 3 = 1). Consider the following rule from the given

grammar for the non-terminal *op*. There are four possible production rules for this non-terminal.

```
(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), the operator /. 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 [20].

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, so 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>` and so on.

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.

4 Experimental Approach

The dataset consists of financial data of 791 non-financial US companies drawn from the S&P Compustat database. The associated S&P overall credit rating for each corporate bond issuer is also obtained from the database.[1]

Of these companies, 57% have an investment rating (AAA, AA, A, or BBB), and 43% have a junk rating. To allow time for the preparation of year-end financial statements, the filing of these statements with the Securities and Exchange Commission (S.E.C), and the development of a bond rating opinion by Standard and Poor rating agency, the bond rating of the company as at 30 April 2000, is matched with financial information drawn from their financial statements as at 31 December 1999. A subset of 600 firms was randomly sampled from the total of 791 firms, to produce two groups of 300 'investment' grade and 300 junk rated firms. The 600 firms were randomly allocated to the training set (420) or the hold-out sample (180), ensuring that each set was equally balanced between investment and non-investment grade ratings.
A total of eight financial variables was selected for inclusion in this study. The selection of these variables was guided both by prior literature in bankruptcy prediction [21, 22, 23], literature on bond rating prediction [14, 24, 25], resulting in an initial judgemental selection of a subset of accounting ratios. These ratios were then further filtered using statistical analysis.

Five groupings of explanatory variables, drawn from financial statements, are given prominence in prior literature as being the prime determinants of bond issue quality and default risk:

i. Liquidity
ii. Debt
iii. Profitability
iv. Activity / Efficiency
v. Size

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 also provides a proxy metric on firm history. The groupings of potential explanatory variables can be represented by a wide range of individual financial ratios, each with slightly differing information content. The groupings themselves 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. Following the examination of a series of financial ratios under each of these headings, the following inputs were selected:

i. Current ratio
ii. Retained earnings to total assets
iii. Interest coverage
iv. Debt ratio
v. Net margin
vi. Market to book value
vii. Log (Total assets)
viii. Return on total assets

The objective in selecting a set of proto-explanatory variables is to choose financial variables that vary between companies in different bond rating classes, and where information overlaps between the variables are minimised. Comparing the means of the above ratios for the two groups of ratings (see table 3), reveals a statistically significant difference between the two groups at both the 5% and the 1% level, and as expected, the financial ratios in each case, for the investment ratings are stronger than those for the junk ratings. The only exception is the current ratio,

[1] S&P is one of the largest credit rating agencies in the world, currently rating about 150,000 issues of securities across 50 countries. It provides credit ratings for about 99.2% of the debt obligations and preferred stock issues which are publicly traded in the US [8].

	Investment grade	**Junk bond**
Current ratio	1.354	1.93
Ret. earn/Tot assets	0.22	-0.12
Interest coverage	7.08	1.21
Debt ratio	0.32	0.53
Net margin	0.07	-0.44
Market to book value	18.52	4.02
Total assets	10083	1876
Return on total assets	0.10	0.04

Table 3: Means of input ratios for investment and junk bond groups of companies.

	CR	RE/TA	IC	DR	NM	MTB	TA	ROA
CR	1	-0.08	-0.01	0.06	-0.27	0.01	-0.18	-0.15
RE/TA	-0.08	1	0.27	-0.64	0.14	0.15	0.15	0.48
IC	-0.01	0.27	1	-0.28	0.06	0.31	0.15	0.41
DR	0.06	-0.64	-0.28	1	-0.05	-0.19	-0.20	-0.27
NM	-0.27	0.14	0.06	-0.05	1	0.01	0.03	0.22
MTB	0.01	0.15	0.31	-0.19	0.01	1	0.04	0.14
TA	-0.18	0.15	0.15	-0.20	0.03	0.04	1	0.07
ROA	-0.15	0.48	0.41	-0.27	0.22	0.14	0.07	1

Table 4: Correlations between financial ratios.

which is stronger for the junk rated companies, possibly indicating a preference for these companies to hoard short-term liquidity, as their access to long-term capital markets is weak. A correlation analysis between the selected ratios (see table 4) indicates that most of the cross-correlations are less than | 0.20 |, with the exception of the debt ratio and (Retained Earnings/Total Assets) ratio pairing, which has a correlation of -0.64.

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 variable-length one-point 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 rulesets (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. The grammar adopted in this study is as follows:

```
<lc> ::= if( <expr> <relop> <expr> )
           class=''Junk'';
         else
           class=''Investment Grade'';

<expr> ::= ( <expr> ) + ( <expr> )
         | <coeff> * <var>

<var> ::= Current_Ratio
        | Retained_Earnings_to_total_assest
        | Interest_Coverage | Debt_Ratio
        | Net_Margin | Market_to_book_value
        | Total_Assets | ln(Total_Assets)
        | Return_on_total_assets

<coeff> ::= ( <coeff> ) <op> ( <coeff> )
          | <float>

<op> ::= + | - | *

<float> ::= 9 | 8 | 7 | 6 | 5 | 4
          | 3 | 2 | 1 | -1 | .1

<relop> ::= <=
```

5 Results

The results from our experiments are now provided. Each of the GE experiments is run for 100 generations, with variable-length, 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. Results

are reported for two population sizes (500 and 1000). To assess the stability of the results across different randomisations of the dataset between training and test data, we recut the dataset five times, maintaining an equal balance of investment and non-investment grade ratings in the resulting training and test datasets.

In our experiments, fitness is defined as the number of correct classifications obtained by an evolved discriminant rule. The results for the best individual of each cut of the dataset, where 30 independent runs were performed for each cut, averaged over all five randomisations of the dataset, for both the 500 and 1000 population sizes, are given in table 5. In each case the overall classification accuracy is provided, and this is then subdivided into the number of true positives N_{tp}, the number of true negatives N_{tn}, and the number of false positives, and false negatives respectively (N_{fp}, N_{fn}).

To assess the overall hit-ratio of the developed models (out-of-sample), Press's Q statistic [26] was calculated for each model. In all cases, the null hypothesis, that the out-of sample classification accuracies are not significantly better than those that could occur by chance alone, was rejected at the 1% level. A t-test of the hit-ratios also rejected a null hypothesis that the classification accuracies were no better than chance at the 1% level. Across all the data recuts, the best individual achieved an 87.56 (84.36)% accuracy in-sample (out-of-sample) when the population size was 500, with the best individual across all data recuts in the population=1000 case obtaining an accuracy of 87.59 (84.92)% accuracy in-sample (out-of-sample). Although the average out-of-sample accuracy obtained for population=1000 slightly exceeds that for population=500, the difference was not found to be statistically significant. A plot of the best and average fitness on each cut of the in-sample dataset, for the population=500 case, can be seen in figure 3, and for case where population=1000 in figure 4.

Examining the structure of the best individual in the case where the initial fitness function was utilised and where population=500 shows that the evolved discriminant function had the following form:

IF (10 + 16 var6 -9 var4 -2 var9) $\geq$ 0 THEN 'Junk' ELSE 'Investment Grade'

where var6 is *Debt Ratio*, var4 is $\frac{Retained\ Earnings}{Total\ Assets}$, and var9 is *Total Assets*.

In the case where population=1000 the best evolved discriminant function had a similar form to the above:

IF (5 + 8 var6 -4 var4 - var9) $\geq$ 0 THEN 'Junk' ELSE 'Investment Grade'

Examining the signs of the coefficients of the evolved rules does not suggest that they conflict with common financial intuition. The rules indicate that low/negative retained earnings, low/negative total assets or high levels of debt finance are symptomatic of a firm that has a junk rating. It is noted that similar risk factors have been identified in predictive models of corporate failure which utilise financial ratios as explanatory inputs [3, 4]. Conversely, low levels of debt, a history of successful profitable trading, and high levels of total assets are symptomatic of firms that have an investment grade rating. Although the two discriminant functions have differing coefficient values, they are in essence very similar, as the differing coefficient values are balanced by the differing constant term which has been evolved in each function.

Considering the individual classification rules, it interesting that despite the potential 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.

5.1 Benchmarking the Results

To provide a benchmark for the results obtained by GE, we compare them with the results obtained on the same recuts of the dataset, using a fully-connected, three-layer, feedforward multilayer perceptron (MLP) trained using the backpropagation algorithm, and with the results obtained using linear discriminant analysis.

The developed MLP networks utilised all the explanatory variables. The optimal number of hidden-layer nodes was found following experimentation on each separate data recut, and varied between two and four nodes. The classification

	Fitness	TP	TN	FP	FN
Train GEBOND500	0.861	185.4	176.4	33.6	24.6
Train GEBOND1000	0.867	183.4	180.8	29.2	26.6
Out-Sample GEBOND500	0.854	77.8	76	14	12.2
Out-Sample GEBOND1000	0.860	78	76.8	13.2	12
Train MLP	0.8690	181.8	183.2	26.8	28.2
Out-sample MLP	0.8500	75.8	77.2	12.8	14.2

Table 5: Performance of the best evolved rules on their training and out-of-sample datasets, averaged over all five randomisations, compared with the classification performance of an MLPs on same datasets.

accuracies for the networks, averaged over all five recuts is provided in table 5.

The levels of classification accuracy obtained with the MLP are competitive with earlier research, with for example [14] obtaining an out-of-sample classification accuracy of approximately 83.3%, although it is noted that the size of the dataset in this study was small. Comparing the results from the MLP with those of GE on the initial fitness function suggests that GE has proven highly competitive with an MLP methodology, producing a similar classification accuracy on the training data, and slightly out-performing the MLP out-of-sample.

Utilising the same dataset recuts as GE, LDA produced results (averaged across all five recuts) of 82.74% in-sample, and 85.22% out-of-sample. Again, GE is competitive against these results in terms of classification accuracy. Comparing the results obtained by the linear classifiers (LDA and GE) against those of an MLP suggests that strong non-linearities between the explanatory variables and the dependent variable are not present.

6 Conclusions & Future Work

In this paper a novel methodology, GE, was described and applied for the purposes of prediction of bond ratings. 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 bond ratings from raw financial information. Despite using data drawn from companies in a variety of industrial sectors, the developed models showed an impressive capability to discriminate between investment and junk rating classifications. The GE-developed models also proved highly competitive with a series of MLP models developed on the same datasets.

Several extensions of the methodology in this study are indicated for future work. One route is the inclusion of non-financial company and industry-level information as input variables. A related possibility would be to concentrate on building rating models for individual industrial sectors. The study can also be extended to encompass the multi-class rating prediction problem. As already noted, there are multiple methodologies available for the generation of classification rules / regression models [27, 28]. Future work could extend this study by examining the general utility of GE vs other methods of generating classification rules, by comparing the performance of a range of methods on a wider range of datasets.

References

[1] O'Neill M. and Ryan C. (2003) *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers 2003.

[2] Bond Market Statistics (2003). New York: The Bond Market Association.

[3] Brabazon, A. and O'Neill. M. (2004). Diagnosing Corporate Stability using Grammatical Evolution, *International Journal of Applied Mathematics and Computer Science*, 14(3), pp. 363-374.

[4] Brabazon, A. and O'Neill, M. (2003). Anticipating Bankruptcy Reorganisation from Raw Financial Data using Grammatical Evo-

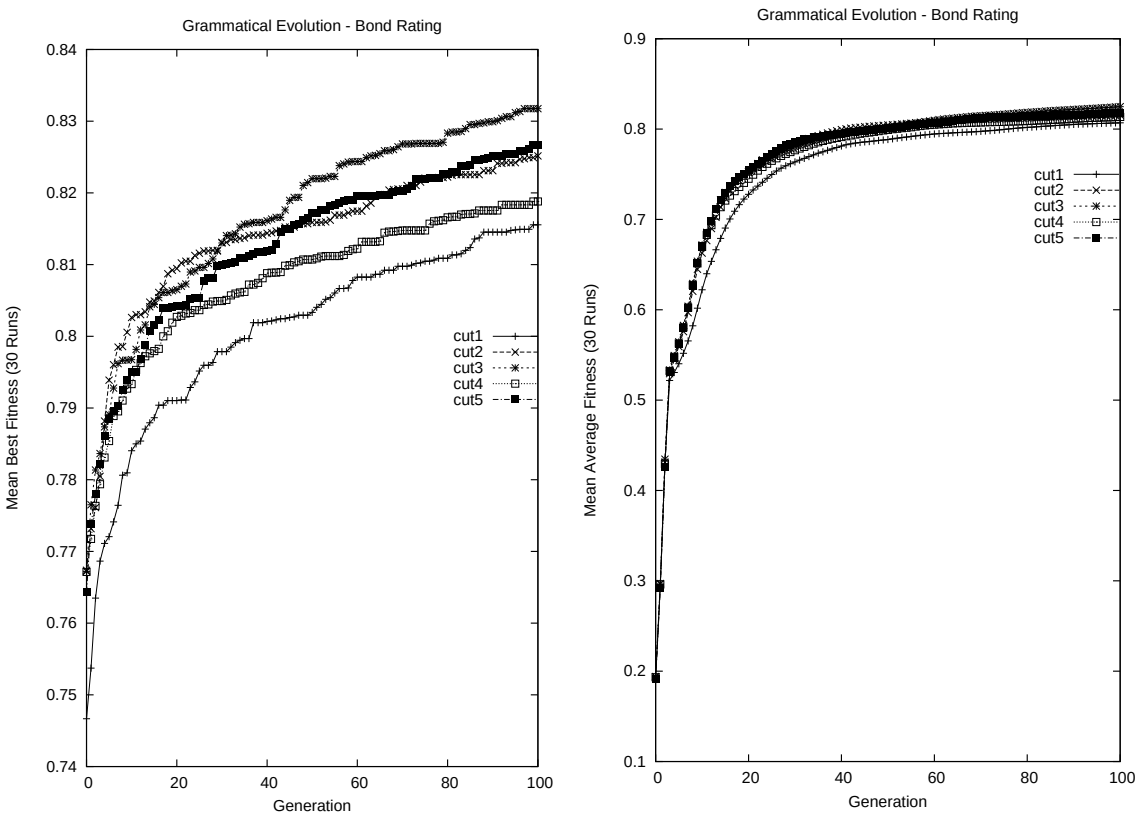

Figure 3: Best and average fitness values of 30 runs on the five recuts of the in-sample dataset with a population size of 500.

lution, Proceedings of EvoIASP 2003, *Lecture Notes in Computer Science (2611): Applications of Evolutionary Computing*, edited by Raidl, G., Meyer, J.A., Middendorf, M., Cagnoni, S., Cardalda, J. J. R., Corne, D. W., Gottlieb, J., Guillot, A., Hart, E., Johnson, C. G., Marchiori, E., pp. 368-378, Berlin: Springer-Verlag.

[5] O'Neill, M. and Brabazon, A. (2004). Grammatical Swarm, in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO 2004*, Lecture Notes in Computer Science (3102), Deb et. al. (eds.), Seattle, USA, June 26-30, 2004, 1, pp. 163-174, Berlin: Springer-Verlag.

[6] Shan, Y., McKay, R., Baxter, R., Abbass, H., Essam, D. and Nguyen, H. (2003). Grammar Model Based Program Evolution, in *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, 1, pp. 478-485, IEEE Press: New Jersey.

[7] Altman, E. (1998). The importance and subtlety of credit rating migration, *Journal of Banking & Finance*, 22, pp. 1231-1247.

[8] Standard and Poor's (2002). Standard and Poor's Rating Services, *Statement at US SEC Public Hearing on the Role and Function of Credit Rating Agencies in the US Securities Markets*, 15 November 2002.

[9] Horrigan, J. (1966). The determination of long term credit standing with financial ratios, *Journal of Accounting Research*, (supplement 1966), pp. 44-62.

[10] Pinches, G. and Mingo, K. (1973). A multivariate analysis of industrial bond ratings, *Journal of Finance*, 28(1), pp. 1-18.

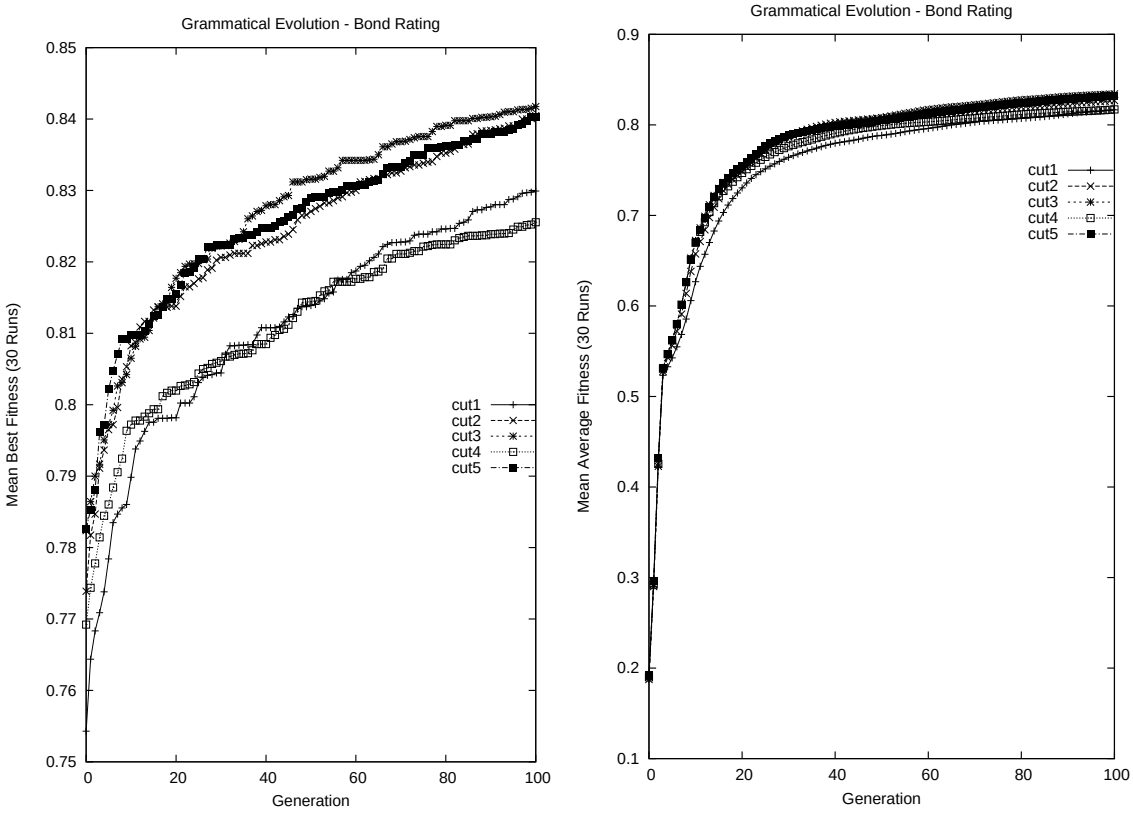

Figure 4: Best and average fitness values of 30 runs on the five recuts of the in-sample dataset with a population size of 1000.

[11] Ederington, H. (1985). Classification models and bond ratings, *Financial Review*, 20(4), pp. 237-262.

[12] Gentry, J., Whitford, D. and Newbold, P. (1988). Predicting industrial bond ratings with a probit model and funds flow components, *Financial Review*, 23(3), pp. 269-286.

[13] Maher, J. and Sen, T. (1997). Predicting bond ratings using neural networks: a comparison with logistic regression, *Intelligent Systems in Accounting, Finance and Management*, 6, pp. 23-40.

[14] Dutta, S. and Shekhar, S. (1988). Bond rating: a non-conservative application of neural networks, *Proceedings of IEEE International Conference on Neural Networks*, II, pp. 443-450.

[15] Fogel, D. (2000). *Evolutionary Computation: Towards a new philosophy of machine intelligence*, New York: IEEE Press.

[16] O'Neill, M. (2001). Automatic Programming in an Arbitrary Language: Evolving Programs in Grammatical Evolution. PhD thesis, University of Limerick, 2001.

[17] O'Neill M. and Ryan C. (2001) Grammatical Evolution, *IEEE Trans. Evolutionary Computation*. 2001.

[18] Ryan C., Collins J.J. and 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, Springer-Verlag.

[19] Koza, J. (1992). *Genetic Programming*. MIT Press.

[20] Lewin, Benjamin. (2000). *Genes VII*. Oxford University Press.

[21] Altman, E. (1993). *Corporate Financial Distress and Bankruptcy*, New York:John Wiley and Sons Inc.

[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] Altman, E. (1968). Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy, *Journal of Finance*, 23, pp. 589-609.

[24] Kamstra, M., Kennedy, P. and Suan, T.K. (2001). Combining Bond Rating Forecasts Using Logit, *The Financial Review* , 37, pp. 75-96.

[25] Singleton, J. and Surkan, A. (1991). Modeling the Judgment of Bond Rating Agencies: Artificial Intelligence Applied to Finance, *Journal of the Midwest Finance Association*, 20, pp. 72-80.

[26] Hair, J., Anderson, R., Tatham, R. and Black, W. (1998). *Multivariate Data Analysis*, Upper Saddle River, New Jersey: Prentice Hall.

[27] Breiman, L., Freidman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*, New York: Chapman and Hall.

[28] Torgo, L. (2000). Partial Linear Trees, in Proceedings of the 17th International Conference on Machine Learning (ICML 2000), Langley, P. (ed), pp. 1007-1014, Morgan Kauffman.