To Create and Compare the Predictive Accuracy of a Genetic Program and an Artificial Neural Network to Predict Company Failure: Final Report

Carl Saptarshi
Student Number: 640032165

April 2017

Contents

1 Background and Introduction

Due to the dynamic and volatile economy that we live in, the number of companies filing for bankruptcy are on the rise, especially during times of economic uncertainty, for example, during a period of recession.
In turn, being able to predict the likelihood of a company failing and filing for bankruptcy is very important and has been a focal point of issue in accounting research and analysis over the past thirty years.

1.1 Background into Bankruptcy

1.1.1 What is Bankruptcy?

When a company (the *debtor*) takes out a loan or borrows money from somewhere else such as a financial institution like a bank (the *creditor*) , it is up to the debtor to ensure that the creditor is repaid the full amount that was borrowed subject to the creditors terms and conditions.

If the debtor starts to fall behind on their payments to the point that they are unable to repay their debts or unable to keep up with the incremental loan repayments, the debtor may file for a Chapter 7 bankruptcy in which the court will appoint a trustee to shut down the company and liquidate their assets, for example, by selling machinery, land and company shares to recover some money which they the trustee can give back to the creditor to clear the company's debt. If the company is still unable to pay back the debt even after this, then they will file for bankruptcy and the company will be terminated. Since the 1960's, as economies have grown, especially in the western world, this problem has been recognised on grander scale in more economically developed countries.

1.1.2 Who does it affect?

Bankruptcy does not just affect those that are employed within that company; it also affects third party members such as shareholders, investors, suppliers, and company clients . CF has been a critical point of focus, especially in the field of financial analysis and for stakeholders who are interested in the performance of the company. Since the 1960's, empirical risk assessment models have been developed which have been used to predict CF. Using current and previous financial data of a company, the likelihood of CF can be predicted for 'n' number of years ahead. This in turn means that loan companies, such as banks, can use this information to determine whether loans should be granted to other firms, knowing the likelihood of a company defaulting or not. This has helped to give banks competitive advantages, as they become aware of how likely a company will be to default, and therefore is able to predict customer behaviour in times of difficulty.
Looking at reports from the American Bankruptcy Institute showed that in the year 2000, 35,742 companies filed for bankruptcy, 43,546 companies in 2008 and 60,837 by 2009, at the peak of the recession. By 2012 this fell to 40,075 and 24,114 by 2016. These statistics clearly indicate the volatility and uncertainty in the economy as it changes, which is part of what makes CF prediction incredibly important, especially for those involved with the company at hand.

1.2 Algorithms for Company Failure prediction

Since the aim of the project to classify whether a company is likely to fail or not, this can be called a binary classification problem which will take in a series of inputs and return a classification which determines whether a company is financially distressed or not. On top of this, a company likely to suffer from financial distress will have certain characteristics associated with them, similarly this would occur for companies not facing this problem. This means that the data should be linearly separable when classifying the data, making this a linear binary classification problem.
There have been several techniques that have been used to predict CF, some of which will be introduced here.

Individual Ratio Selection (IRS) -In the 1960's, Beaver introduced IRS. This process involved selecting thirty financial variables, converting these to ratios. Based on a certain threshold for each variable, this would determine if a company is likely to fail or not.

Multivariate Discriminant Analysis (MDA) - Altman created this technique in the 1960's, which takes uses a discriminant function to score a company. This function uses five financially weighted ratios and based on the overall discriminant score, the company can be classified.

Genetic Programming (GP) - GP's are inspired by Darwin's Theory of Evolution, used for prediction and classification. A population of functions is created and fitter individuals in the population are more likely to survive and produce offspring that are even more suited to the environment. The aim is to create an optimal function which can give the most accurate prediction in terms of classification accuracy.

Artificial Neural Networks (ANN) - This technique is inspired by the interconnectivity of the brain and applies this to prediction and classification. ANN's are made of an input layer, hidden layers and the output layer which outputs the classification based on the input. The network uses the weights on the synapses of the nodes that connect one node to the next, which are tweaked to allow the network to learn and give a more accurate classification for unknown datum.

Supervised Learning - GP's and ANN's fall under the umbrella of Machine Learning (ML). ML is a form of Artificial Intelligence (AI) that allows a computer program to learn without the use of explicit programming. This means whilst the program is running, it will start to form patterns based on the data that is fed in, and adapt the program appropriately to try to produce the best results until a termination criteria is met. In supervised learning, the output after each iteration of each algorithm is compared against the already known desired output. This can then be used for comparison to check whether or not the programs have correctly classified a company financially distressed or not. For every iteration, the program will start to learn, so the accuracy of the classifications will start to increase over time. Eventually, when an unknown set of data is inputted into the programs, the program will be able to correctly produce an output to declare if the company in question will likely to fail or not along with a percentage of certainty.

1.3 Motivations For this Project

2 Summary of literature review and specification

2.1 Literature Review

All the techniques mentioned in section 1.2 have been used extensively in the classification and prediction of problems. Altman and Ohlson, who are the pioneers of CF prediction since the 1960's used selected financial variables from multiple companies bank statements to predict CF. These variables (*key performance indicators* (KPI's)) were used as they believed were important factors that indicated whether a company was financially distressed. To make companies more comparable, both techniques involved converting the KPI's into ratios as a method of standardising the data. Due to the success of both of their methods, newer techniques proposed are based around their financial ratios and use the prediction accuracy for both methods as benchmarks to compare their new proposed work against techniques already in use.

Overall, Altman's technique was superior to Beaver's method, but only if the KPI's were jointly distributed according to a multivariate normal distribution. Otherwise, it was prone to errors. However, the MDA technique was favoured as it could use multivariate data at once to get an overall prediction rather than taking each ratio and scoring that to give predictions.
Wilson and Lensburg took modern approaches by implementing ANN's and GP's to this classification problem. Both techniques can handle noisy data that is not normally distributed better than Altman's MDA, showing that both ANN's and GP's have potential to be more accurate than IRS and MDA.

ANN's tend to perform very well in terms of efficiency and accuracy. Wilson used an ANN approach with a 5 10 2 structure. To improve accuracy, the Monte-Carlo technique was used to give a better representation of predicative accuracy. Overall, they achieved a 97.5% accuracy on their testing dataset, making this much more accurate than MDA and IRS.
ANN's suffer from being unable to produce a readable function to indicate how it came to the solution, other than viewing the synaptic weights. Another issue faced is that since a function is unable to be produced, the ANN is unable to tell us what influence each KPI has when predicting CF.
GP's can overcome some of the problems of an ANN as they are able to produce a user readable function. It is possible to see what kind of influence each KPI has relative to each other. Lee used a decision tree (DT) method to predict CF. Lee used eight different KPI's when approaching this problem. Using this GP method, the testing accuracy of 92.91%. Rostamy used a similar approach to Lee, using five different KPI's. After training the GP, it could correctly predict if a company would fail 90% of the time, which was like MDA, however more flexibile in terms of what type of data it could accept.
GP's may work slower as they explore a large search space and may be restricted to certain limitations e.g. a maximum tree depth. For each crossover and mutation, the depth of tree may increase. This can increase the computation time rapidly. When designing and implementing the GP, these factors will typically be accounted for as seen in Etemadi's et al paper.

Through the research completed, many papers used Altman's KPI's as their inputs. However, as Altman suggested, these ratios may not necessarily be the most optimal, but these still provided the best alternative discriminant function to work with at the time. Since then economies have changed significantly, these ratios may not necessarily be the best to use to predict CF, but may

still be significant enough to give an accurate prediction. As seen by Back, Rostamy and Lee, other ratios have been used to predict CF, and achieved similar results to MDA, which could potentially be more significant now. Wilson used Altman's KPI's and achieved the 97.5% accuracy, with far fewer ratios relative to Back and Lee, which must be taken into consideration.

2.2 Project Specification

After careful consideration of the researched techniques, I will be predicting CF using ANN's and GP's due to their strong accuracy rates and ability to handle noisy data. though they do have drawbacks, I will aim to minimise these through the project specification and implementation.

A dataset of 134 different Small-Medium Enterprises (SME's) were collected, using between three to five years' worth of data for each, giving a total of approximately 700 data records. The decision to use SME's was to have more consistent, comparable data, as shown by Altman.The spreadsheet consists of financial statement names and variables which can be used for the programs that will be made. To begin with, I will use Altman's KPI's as these have proven to be successful even today. The data provided also shows whether or not a company has failed not failed during a certain time period. Since there are KPI's which will be used, and a clear binary output (0 or 1) which is known and will be used, this type of ML is known as a supervised learning task.

Two programs will be created for this project. Firstly, a feed forward ANN with back propagation and secondly a regression tree GP will be created. The reasons that I have chosen to use these are due to the research that I have undertaken, ANN's have been known to produce very accurate results in an efficient time manner. The reason that GP is a valid technique to use for this task is that GP's can be used in prediction, but also they are able to create a function that will directly map the input to the output in order to produce the given result.
For the ANN, I will use a library - *sklearn*, a module in the Python programming language, as this library consists of optimised mathematical functions which will enable the creation of the ANN to be much easier due to possible time constraints, and because this will used as a benchmark to test against the GP. Although sklearn does provide support for regression trees in Python, I will make it from scratch, as this will provide me with more flexibility to manipulate the functions more easily towards the problem at hand. Once The GP and ANN have been created, run, and the results tabulated and graphed, I will then be able to test each one independently and against each other. They will also be compared to Altman's benchmarks to compare the predictive accuracies from the two programs I will be making.

The task is also known as a binary classification problem. This means that the output of each of the programs determine whether a company can be classified as likely to fail or will not fail. To represent this classification, the value of 0 will represent a non-distressed company, and a 1 will represent a financially distressed company. Since this representation will be a number, in order to break this down further, the actual floating point value of the output (which will be between 0 and 1) will be used to represent the likelihood of failure as a probability. For example, if the value was 0.618, it could be said that the company has a probability of 0.312 of staying afloat for the forthcoming year. Whereas if the value was 0.111 then the company has a 0.899 probability of staying afloat for the forthcoming year. Here, a clear differentiation can be made between two companies, one which is more likely to fail than the other.

3 Design

For this task, two different models had to be designed and created as stated in section 2.2. Due to the size of this project, by breaking each of the models down into smaller, more manageable parts, it gave a better, clearer structure to how the two models would eventually be implemented through code.

3.1 Data Collection

Before staring the design of the GP and the ANN, the first ting to do was to collect all the data that I planned to use. The data that collected had been given to me by the University of Exeter Business School as they had easy access to multiple years worth of data for hundreds of companies. Due to the fact that each economy is different financially, and companies may perform better or worse in different economies, to make the data more comparable, the data that I collected was solely from American companies from the US economy.
The data collected contained 671 rows of data, with 31 different variables. For this project, only eight of the 31 variables were financial variables that were indicators of a company's performance. For example, *net income* and *sales* would be considered to be variables that could affect the company performance, however variables like *company name* and *ticket*, would be much less likely to affect the company's performance. Therefore, to make the data more usable, variables not considered to be KPI variables were removed from the dataset. After completing this, the next step was to remove incomplete data. Some of the rows of the data contained question marks (?) where cells were missing data. As this could have potentially affected the programs and how the data is read in and make the data harder to work with, any rows with any incomplete data were filtered out. Now the dataset contained only complete data with the key financial variables. Due to the types of the companies being given, to make smaller companies more comparable to larger companies, the next step was to standardise these variables into five ratios, similar to the methods that Altman used to standardise his dataset. This new standardised dataset contained five variables, labelled X1, X2, X3, X4, X5, each representing a specific ratio, and finally the classification for that particular row, to indicate whether or not that particular company had failed or not, indicated by a 1 for failed, and a 0 for not failed. Since this dataset was now in the right format, the next stage was to design the GP and the ANN that I chose implement.

3.2 Artificial Neural Network Design

Now that the data collection process had been completed, the next step was to design the ANN. After completing the relevant research, the type of ANN I chose to implement was a feed forward artificial neural network. The reason that this was chosen is that baed on the research that was undertaken, the papers that implemented an ANN used a simple feed forward network as this avoided any unnecessary complexity that would occur if other types of ANN's were used. An ANN usually follows the same layout which makes them applicable to an array of complex problems. The basic structure of the ANN consisted of an input layer, hidden layer(s) and an output layer.

3.2.1 Input Layer and Synaptic Weights

Now that the dataset had been modified into five KPI's and a classification feature, I had to decide what the input features would be used for this network. Since ANN's are able to cope with high

dimensional, non linear data, I decided to use all five of the KPI ratios that I had created from the dataset as the ANN would be more than capable to handle all the data that I was planning on giving it.
Due to the stochastic nature of an ANN, when initialising the network before the data from the input features are read into the network, the weights that each node in the previous layer to each node in the next layer also needed to be initialised. As part of the deign, I chose to completely randomise the weights on each of the synapses. If the weights were not initialised randomly, when the network starts to learn, it would learn in the exactly same way every time the program is run because the network will always start at the same position in the search space, which means it will always follow the same route to get to a solution which would always be the same since the stochastic element has been removed. By initialising the weights randomly, the symmetry of the network is broken, which allows the network to be initialised differently, so the weighted input signal that moves from one layer to the next would always be different, allowing more of the search space to be explored, which would allow more solutions (possibly more optimal solutions) to be found.

3.2.2 Hidden Layers and Activation Function

The next step is the structure of the hidden layers of the network. For this, I have decided to use two hidden layers with between 1-100 nodes. Through research, it was found that a sufficiently wide ANN with just one hidden layer would tend to be enough to train the ANN without memorising on the network. However, the wider the network, the more likely it is that the network will be to memorise the data, which means generalisation will be poorer, such that although the data has learned well when training the network, the accuracy when testing the network would be poor as the network has overfitted the data. Therefore to attempt to avoid this, I will use two hidden layers as this would allow the network to generalise better without network memorisation. Rather than using trial and error, I decided to use cross validation to find the most optimal number of nodes for each one of the hidden layers. To do this, the network would be cross-validated 5 times. The reason cross validation works is that it will attempt to perform all the permutations of the nodes on the network with the intension of finding 'n' number of nodes in the first hidden layer and 'm' number of nodes in the second layer which would result in the best training accuracy. Once these have been found, this network can then be tested on by using the testing set of data.

At every hidden layer node, the weighted sum of the input to that node will be calculated. This weighted sum will then be passed through an activation function which will then send the output of this node to be the input to the next layers nodes. For the simple ANN that is being designed, I will initially use a sigmoid activation function. The reason this is a good idea and should be part of the design is that the sigmoid function adds an element of non-linearity to the to the model, which allows the computation of nontrivial problems using only a small number of nodes, which is what makes this particular function much more popular relative to other activation functions like a step function.

3.2.3 Output Layer and Learning Mechanism

After the input has been fed forward through the network, the output will be produced. When training the network, I will classify the outputs as either a 0 or 1 by passing the raw value through another activation function. If the value outputted is below 0.5, then it will be classified as a 0,

otherwise it would be classified as a 1. Using this, the error percentage can then be measured against the true results, and the weights will be altered accordingly using a learning mechanism. For this network, through the research conducted, the back - propagation learning method should be sufficient enough to allow the network to learn, to try to minimise the error rate, to give the best possible accuracy, showing that the network has trained.
Once the ANN has produced an accuracy of 80% or higher on the training dataset, I will then use a testing dataset to test the predictive accuracy of the classifier on unknown data. This process will occur multiple times during the process of cross validation as the network will be selecting various weights and number of nodes with the aim of producing the most optimal network within a sufficient amount of time.

3.3 Genetic Program Design

3.3.1 Generating a Population

3.3.2 Selection

3.3.3 Crossover and Mutation

3.3.4 Termination Criteria

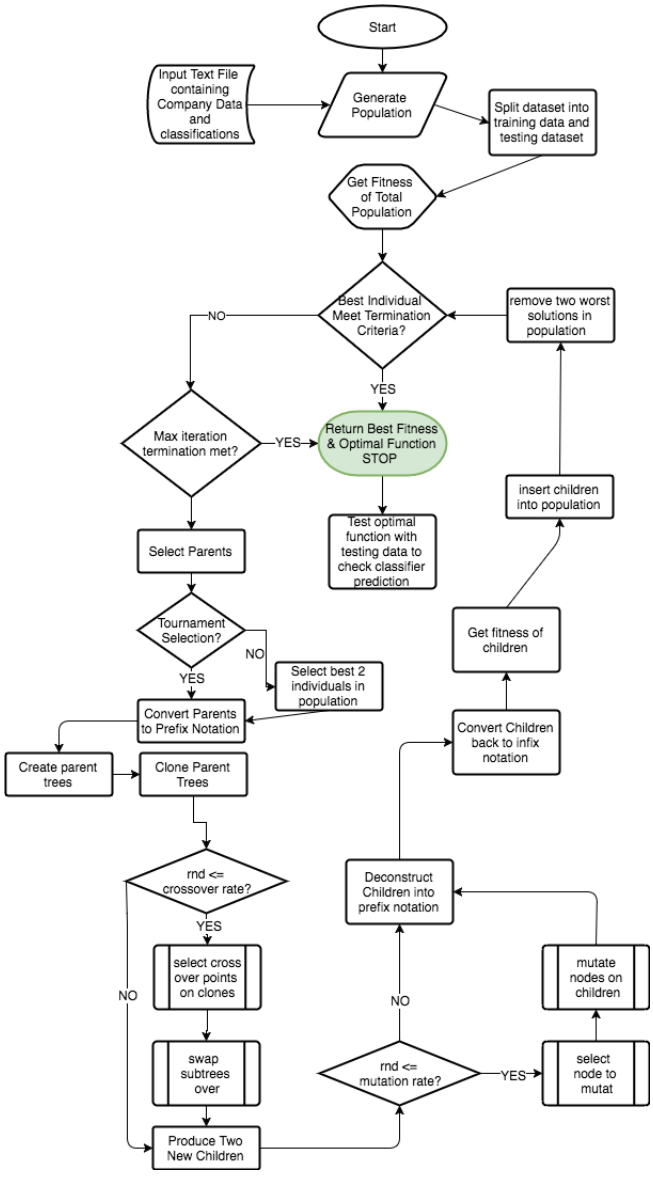

Figure 1: Figure1: Flow chart to represent the Genetic Program that has been designed

3.4 Choice of Programming Language

4 Development

5 Testing

6 Description of the final product

7 Evaluation of the final product

8 Critical assessment of the project

9 Future Work

10 Conclusion