Expert Systems with Applications 36 (2009) 9065–9071

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Are we modelling the right thing? The impact of incorrect problem specification in credit scoring

Steven Finlay *

Department of Management Science, Lancaster University Management School, Lancaster University, Lancaster LA1 HYW, United Kingdom

ARTICLE INFO

Keywords:
Genetic algorithms
Credit scoring
Forecasting

ABSTRACT

Classification and regression models are widely used by mainstream credit granting institutions to assess the risk of customer default. In practice, the objectives used to derive model parameters and the business objectives used to assess models differ. Models parameters are determined by minimising some function or error or by maximising likelihood, but performance is assessed using global measures such as the GINI coefficient, or the misclassification rate at a specific point in the score distribution. This paper seeks to determine the impact on performance that results from having different objectives for model construction and model assessment. To do this a genetic algorithm (GA) is utilized to generate linear scoring models that directly optimise business measures of interest. The performance of the GA models is then compared to those constructed using logistic and linear regression. Empirical results show that all models perform similarly well, suggesting that modelling and business objectives are well aligned.

1. Introduction

All mainstream credit granting institutions use credit scoring – mechanically derived forecasting models of customer behaviour – to make decisions about whom to extend credit to and on what terms. The most widely used credit scoring models predict a simple binary outcome; that is, the likelihood that an individual will be a 'good' customer who repays the credit advanced to them, or a 'bad' customer who defaults. Despite much research into the applicability of a wide variety of classification and regression methods to credit scoring problems, logistic regression remains the most widely used method in practice (Crook, Edelman, & Thomas, 2007; Finlay, 2008). This is mainly attributed to the fact that logistic regression produces simple models that are easily interpretable, as well as empirical evidence suggesting that the performance of simple linear models is only fractionally worse than more complex model forms such as neural networks and support vector machines (Baesens et al., 2003).

In many real world situations, the objective a lender is trying to optimise through the use of a credit scoring model is different from the objective used during model development. Therefore, a key question – that has not been widely considered by the credit scoring community – is; are we modelling the right thing? And if not, what is impact of not doing so? As a simple illustration, consider logistic regression applied to a binary classification problem, where the dependent variable, y, takes values of 0 or 1. Through the application of an appropriate algorithm, a model is derived that maximises likelihood over the set of n observed cases:

$$\prod_{i=1}^{n}\left(P_i^{y_i}+(1-P_i)^{(1-y_i)}\right) \tag{1}$$

where P_i is the posterior probability that $y_i = 1$, calculated as a function of independent variables. Yet, for many practitioners the actual point estimate for an observation is of little interest. What is of primary importance is the relative performance at specific points in the distribution of ranked model scores (Thomas, Banasik, & Crook, 2001). It is also true that for some decisions (such as where a fixed accept rate policy is in operation) the only concern is that observations fall on the correct side of the decision rule applied. Whether, an individual only just passes the cut-off score or exceeds it by a great margin is irrelevant (Hand, 2005). This can be demonstrated by considering a hypothetical example. Imagine that there exist two models that generate probabilistic estimates of credit applications being good credit risks. Two credit applications are scored by each model to produce the results shown in Table 1.

Now assume that both cases are revealed to be good payers. From a maximum likelihood perspective, Model 1 outperforms Model 2. Yet, if a lender was using these models to make credit granting decisions, say on the basis of accepting only those where the estimated probability of being good exceeds 0.8, then Model 2 is better because both cases would be accepted. Maximising likelihood is therefore no guarantee of optimal model performance in this case.

A Genetic Algorithm (GA) is a data driven, non-parametric heuristic search process, where the training algorithm can be chosen to

* Tel.: +44 1772 798673.
E-mail address: steven.finlay@btinternet.com

0957-4174/$ - see front matter Published by Elsevier Ltd.
doi:10.1016/j.eswa.2008.12.016

Table 1
Hypothetical model estimates.

Model	Model estimates		Likelihood (from Eq. (1))
	$P(y_1 = \text{good})$	$P(y_2 = \text{good})$	
Model 1	0.75	0.98	0.735
Model 2	0.82	0.85	0.697

optimise a wide range of objective functions. Because the training algorithm is guided only by the performance of competing solutions, GAs have the potential to generate models that outperform other approaches to credit scoring in situations where the objective function that a user wishes to optimise, differs from that used within the modelling process.

Previous studies where GAs have been used to develop credit scoring models have reported mixed findings. Fogarty and Ireson (1993/4) took a sample of over fifty thousand accepted credit card applications and compared a GA derived Bayesian classifier with decision rules derived from a number of techniques including a nearest neighbour clustering algorithm, a decision tree and a simple Bayesian classifier. They found that the GA derived classifier performed better than other methods when assessed on classification rates, but did not perform better than a simple decision rule to classify all cases as good. Desai, Conway, Crook, and Overstreet (1997) looked at a three-way classification problem where accounts were classified as good, poor or bad payers. They reported that a GA approach was marginally better at classifying the worst accounts (bad payers) than linear discriminant analysis, logistic regression and a variety of neural network models, but did not perform as well when measured in terms of classification performance on good and poor paying accounts. Yobas, Crook, and Ross (2000) reported that while a GA derived model performed better than neural networks and decision trees on the development sample (no validation sample performance was available for the GA derived model), all three methods were outperformed by linear discriminant analysis. While the results and methodologies applied in these previous studies differ, one feature that they all have in common is that they only considered misclassification performance metrics for which the non-GA approaches used in the study were generally known to provide good levels of performance. It is, therefore, no surprise that a GA approach was not found to significantly outperform the alternative model development approaches examined.

In this paper, a GA approach is again explored, but incorporating a number of features that differentiate it from previous studies. First, the objective is primarily to determine the sensitivity of models developed using standard approaches to differences between modelling and business objectives. The actual performance of GA derived models is only a secondary consideration. Second, rather than simply judging performance of competing models on the basis of a single misclassification measure, model performance is assessed using several different criteria:

- The maximisation of the GINI coefficient (a measure of the area under the receiver operator curve) which for a discrete population of n observations that fall into one of two classes and ranked by model score, can be calculated using the Brown formula:

$$1 - \sum_{i=2}^{n} [G(i) + G(i-1)][B(i) - B(i-1)] \tag{2}$$

  where G and B represent the cumulative proportion of cases falling into each class respectively.
- The minimisation of the proportion of bads within the highest scoring x% of the population; that is, the number of bads scoring $\geqslant c$, where c is the cut-off score at or above x% of the population score. For the purposes of this study values of x of 5, 10, 25 and 50 percent were considered.

In each case a GA is applied to generate a scoring model that maximised each objective independently, whereas a single competing model was constructed and assessed using the competitor approaches. Second, two large real world data sets are used, whereas previous studies have been based on relatively low dimensional data sets and small samples (with the exception of Fogarty and Ireson's study). Third, solutions are considered with and without seeding – the process whereby a genetic algorithm is initialised using a number of pre-existing solutions found using some alternative technique. The GA is then applied in an attempt to improve upon the performance of the original seed solution(s).

Empirical results are presented for the two data sets; with the performance of the GA derived models compared to models constructed using logistic regression and multiple OLS regression.

2. Overview of genetic algorithms

The theory of GAs was developed in the late 1960s and early 1970s by John Holland and his associates as a means to study evolutionary processes in nature (Holland, 1975), but they were quickly adopted as a heuristic approach applicable to a wide range of optimisation problems (De Jong, 1975; Hollstien, 1971). The general principles of GAs are analogous to Darwinian principles of natural selection and survival of the fittest, and the terminology employed to describe GA training and selection is taken from the biological analogy.

With GAs, a set of possible solutions to a given problem is analogous to a population of individuals in the natural world. The goal is to combine together and mutate different solutions so that over time fitter (better) solutions evolve. Each individual solution within the population is represented in the form of a finite length string, comprising a finite alphabet where the string and its component characters are analogous to chromosomes and genes, respectively (Goldberg, 1989a). From an initial (usually randomly generated) population of strings, new populations are created over a number of generations (iterations) through the application of the following genetic operators:

- Selection: from the existing population, a number of strings are selected for breeding, with selection favouring those strings that represent the best solutions found to date.
- Crossover: from the selected population, pairs of strings are matched for breeding. 'Child' strings are created by selecting and combining different characters from each of the parent strings.
- Mutation: each character within a string has the chance to undergo mutation, based on some random process. If selected, then the value of the character within the string is randomly reassigned to one of the possible values defined by the encoding alphabet.

The algorithm terminates when a given number of generations have occurred, or when the improvement from one generation to the next falls below a specified threshold. Despite the relative simplicity of genetic algorithms, they have been successfully applied to a wide range of diverse and complex optimisation problems (Coley, 1999; Mitchell, 1996).

3. Design and implementation of genetic algorithms

A number of parameters need to be selected for GA training, and as with methods such as neural networks, the parameters that