Expert Systems with Applications 42 (2015) 2998–3004

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Genetic algorithms for credit scoring: Alternative fitness function performance comparison

Vaclav Kozeny

Department of Economics and Development, Faculty of Tropical AgriSciences, Czech University of Life Sciences Prague, Czech Republic

ARTICLE INFO

Article history:
Available online 26 November 2014

Keywords:
Genetic algorithms
Credit scoring
Evolutionary techniques
Fitness function
Finance
Risk management

ABSTRACT

Credit scoring methods have been widely investigated by researchers; recently, genetic algorithms have attracted particular attention. Many research papers comparing the performance of genetic algorithms and traditional scoring techniques have been published, but most do not provide enough detail about the fitness function used by the genetic algorithm—despite the fact that fitness function has a key influence on the model's overall performance. The aim of this paper is to evaluate the predictive performance of different fitness functions used by genetic algorithms in credit scoring. An alternative fitness function based on a variable bitmask is proposed, and its performance then compared with fitness functions based on a polynomial equation as well as an estimation of parameter range. The results suggest that the bitmask is superior to the two other methods in both accuracy and sensitivity. The Wilcoxon matched-pairs sign rank test and paired t-Test indicate these results are statistically significant.

1. Introduction

Extending credit to the public is a core business of banks worldwide; the primary decision they face is whether to grant a loan to a potential customer. It is therefore essential that financial institutions are able to accurately differentiate between good and bad payers: this ability is limited by the data available to the bank at the time of application screening. Various credit-scoring methods have been developed to assist with this process. The most common ones, based on logistic regression, linear discriminant analysis, or k-Nearest Neighbor, are summarized by Vojtek and Kocenda (2006). In consumer lending, scoring methods draw largely on socio-demographic characteristics provided by clients in their loan application form. In their study, Avery, Calem, and Canner (2004) demonstrated that besides this rather static information, dynamic events in an individual's life can have a significant impact on their credit worthiness. Unfortunately, this type of information is hard to obtain. Individual default risk is also important from a regulatory perspective, as it contributes to the portfolio risk of the bank which is monitored by supervisory bodies. However, a simple addition of these risks may not be the best indicator of the total portfolio risk. In this context Jacobson and Roszbach (2003) proposed a method of weighting individual default risk estimates and applying them to the portfolio valuation model based on value-at-risk.

Generally in lending practice it is not sufficient to have a scorecard developed as it needs to be constantly validated as the market and demographic conditions change. Scorecard development and validation has been the focus of various studies (Dinh & Kleimeier, 2007; Lopez & Saidenberg, 2000; Wu & Olson, 2010). Furthermore, macroeconomic conditions usually influence the bank's overall lending policy as they have a global influence on market conditions (Bonfim, 2009; Stiglitz & Weiss, 1981).

Given the importance of credit scoring and its potential impact on a bank's business, it is unsurprising that traditional ways of assessing the credit worthiness of individuals are constantly being updated. Numerous studies comparing the performance of traditional and modern methods have been, and are being, conducted. For example, a comprehensive comparison of machine learning models with a traditional expert system was published by Ben-David and Frank (2009).

Biologically inspired techniques such as neural networks and genetic algorithms (GA) are becoming increasingly popular: their predictive power in credit scoring is being researched and compared with traditional models. Some studies indicate that these techniques can produce more accurate predictions (Desai, Conway, Crook, & Overstreet, 1997) than traditional approaches but other studies suggest they are less accurate (Fogarty & Ireson, 1993) or report mixed results (Desai, Crook, & Overstreet, 1996; Finlay, 2009). A review of the current state-of-the-art approaches to financial distress definition and prediction modeling was published by Sun, Li, Huang, and He (2014). A concise summary of the research conducted during the last decade in the field of evolutionary computing with its application to credit scoring has been published by Marques, Garcia, and Sanchez (2013).

http://dx.doi.org/10.1016/j.eswa.2014.11.028
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

Genetic algorithms were first introduced by Holland (1975) as an abstraction of biological evolution. A genetic algorithm uses genetic inspired operators to evolve an initial population into a new population. Each population comprises of chromosomes that represent genetically encoded individual solutions to a specific problem. Each individual has a fitness score assigned to them, which represents its ability in terms of a solution. A new population is evolved by using operators of crossover, mutation, and selection, where selection is based on the individual's fitness and influences its ability to reproduce into the next generation. Detailed information about different genetic operators, their functions and usage can be found in Mitchell (1998) or Michalewicz (1996).

The performance of genetic algorithms depends to a large degree on the parameters which are under the control of the researcher, requiring adjustments to deal with the specific problem at hand. These parameters and namely the fitness function therefore have to be carefully selected to match the specifics of credit scoring.

In current credit scoring research, GAs have been used in two different ways. The first area of application is a hybrid approach in which GAs are being used with other methods such as neural networks. In their research, Sustersic, Mramor, and Zupan (2009) use GAs to preselect the variables to be used by neural networks and logistic regression to develop a scoring model. Similarly, Chi and Hsu (2012) use GAs to preselect variables for their dual scoring model construction. This model comprises of both the credit bureau scoring model and the bank's own scoring model. Oreski, Oreski, and Oreski (2012) used a combination of GAs and neural networks to preselect variables and subsequently build a scoring model. Oreski and Oreski (2014) build on their previous research of GAs, and neural networks. They propose a method of incorporating feature selection into the GAs which provides a higher fitness starting population and faster convergence to optimum solution. Chen and Huang (2003) developed a scoring model using neural networks and then used GAs to provide more insight into the group of rejected applicants by conditional reclassification. An application of GAs to estimate validity constraints for the case-based reasoning model is presented by Vukovic, Delibasic, Uzelac, and Suknovic (2012).

The second area of application is the use of GAs as a complete standalone method. Gordini (2014) used genetic algorithms to generate classification rules for SME bankruptcy prediction. Competitive results have been achieved by Finlay (2009), who compared the performance of logistic and linear regression with GAs using a linear fitness function. Most literature, however, does not give enough detail as to the type of fitness function used. A description of a polynomial fitness function can be found in Thomas, Edelman, and Crook (2002). Another approach was proposed by Yobas, Crook, and Ross (2000), who used an estimation of parameter ranges as a fitness function.

These experiments were conducted using different data samples under different conditions. To the best of the author's knowledge, no study has been published comparing different approaches to fitness function selection using the same dataset.

It is the aim of this paper to propose an alternative fitness function based on a variable bitmask, investigate its performance, and compare it with the predictive ability of GAs using a polynomial fitness function, and with GAs using variable range estimation fitness function.

2. Materials and methods

Credit scoring can be described as a classification problem. Traditionally clients have been classified into two groups—*good* and *bad*. This paper adopts the traditional approach but alternative approaches are also possible. A study classifying clients into three groups—good, poor and bad— has been published by Desai et al. (1997). Different studies propose methods to additionally reclassify the rejected groups (Chen & Huang, 2003; Chuang & Lin, 2009; Kim & Sohn, 2004). Some researchers claim that clients should be classified based on profit or net present value they bring to the bank. For example, Finlay (2010) uses GAs to construct profit maximizing scoring models, Blöchlinger and Leippold (2006) investigate ROC curves of scoring models with the aim of deriving a profit maximizing cut-off while Dionne, Artís, and Guillén (1996) extend traditional scoring model by inclusion of profit assessment based on collections costs.

The two-way classification problem can be described formally as follows:

Each customer x is classified by D variables $x = (x_1, x_2, \ldots, x_D)$, where each variable is of range V_j; $j \leqslant D$. The input feature space is then $V = \prod_{j=1}^{n} V_j = \{(x_1, \ldots, x_D) | x_j \in V_j\}$. A chromosome represents a mapping (scoring) function $f: V \rightarrow \{good, bad\}$ that predicts the type of a new credit applicant. The real observed client status in the sample is denoted as $y \in \{good, bad\}$. The fitness function is a combination of the mapping function and its corresponding fitness score. The training of the GA is performed on a client sample S with known characteristics and status:

$$S = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\} \tag{1}$$

where $\vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ is a client and y_i his corresponding status. The fitness score Φ is represented by accuracy calculated as the number of correct predictions divided by the total number of cases in sample N:

$$\Phi(f) = \#\{i \leqslant N | f(\vec{x}_i) = y_i\} / N \tag{2}$$

In this paper three main definitions of fitness function have been used: a fitness function defined by a polynomial equation, a fitness function using range estimates of each independent variable, and a fitness function based on a bitmask for every independent variable covering any combination of its possible values.

Since the paper focuses on fitness functions, to ensure comparability all models had key genetic operators set equally. Each of the genetic operators was fixed after experimentation with its alternatives. A final selection was made based on performance under the given technical constraints. Key characteristics were the ability to consistently reach higher optima solutions and the necessary time to do so.

Each model was initiated with the creation of an initial population of 200 chromosomes. The length of each chromosome in genes was dependent on the type of model as explained in Sections 2.1–2.3. The polynomial model had 23 genes, the range model had 30 genes and the bitmask model had 33 genes in each chromosome. After the model initiation a series of steps was carried out repeatedly. First, the fitness was calculated for every individual solution (chromosome) in the population. Subsequently all chromosomes were ranked based on their respective fitness scores and the elite 5% were copied unchanged to the next population. Additionally, forward migration was used copying 20% of the best fitness chromosomes to the next generation automatically every 20 generations. The third step was to select part of the population for crossover. Stochastic uniform sampling was applied as the selection method. This approach is similar to the popular roulette wheel selection method. The wheel can be constructed in various ways: one of the most frequent models used is fitness proportionate. In this case the wheel is divided into m sections where m equals the number of chromosomes in population. Each section then represents one chromosome; the size of the section is equal to its fitness. In this way, solutions with higher fitness have a greater chance of being selected for reproduction. An example of a roulette wheel with four chromosomes in population is shown in Fig. 1.

In order to select four parents from Fig. 1, the roulette wheel has to be spun four times. With each spin all represented chromosomes have a constant probability of being selected. The first chromosome has a selection probability 25% while the second chromosome 50%, the third 15% and the fourth 10%. The probability of being selected p_i of each chromosome depends on its fitness score $\Phi(f_i)$ and is defined as:

$$p_i = \Phi(f_i) \Big/ \sum_{i=1}^{m} \Phi(f_i) \tag{3}$$

It can be observed that when there is a solution present which has a significantly higher fitness than the rest of the population, this can lead to a quick convergence to such a solution. Stochastic universal sampling builds on the roulette wheel approach, but it yields better population diversity by modifying the approach outlined above. Instead of spinning the wheel n times against one roulette ball, the wheel is spun once against an equidistant net of n balls. These n balls point to n sections, i.e. to n chromosomes, which are selected for the next generation or for the further procedure. As they are selected, the chromosomes are paired to form a group of parents. Offspring were formed in three ways: either by copying the parents unchanged into the next generation population, by applying mutation or by applying a crossover operator with probability p_c = 80%. The applied crossover technique is single point. In single point crossover the same random point in both parent chromosomes z_1 and z_2 is chosen. Chromosomes z_1 and z_2 are then split into two parts. The part before the random split point is z_{1a} and z_{2a} respectively and the part behind the split point z_{1b} and z_{2b}. Offspring chromosomes z'_1 and z'_2 are formed by exchanging the parts behind the split point and are equal to $z_{1a}\,z_{2b}$ and $z_{1b}\,z_{2a}$ respectively (Fig. 2).

Elitism and forward migration were used in order to ensure the best fitness chromosomes will not be destroyed by crossover or mutation. Unfortunately, this leads to a less diverse population, which in turn may lead us to local optima solution. In order to ensure diversity in the population, a relatively higher probability of mutation has been selected. Mutation was carried out to form remaining offspring with the probability of a gene being mutated p_m = 15%.

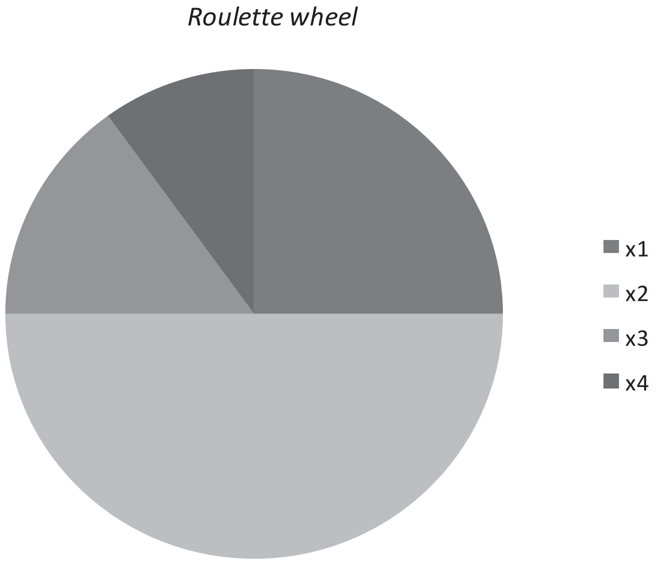

Fig. 1. Roulette wheel.

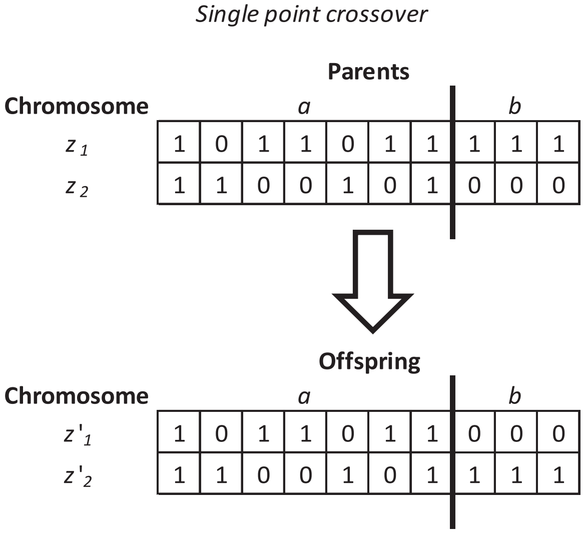

Fig. 2. Single point crossover.

All chromosomes were constructed as binary strings although they were stored as decimal numbers to be converted back into binary strings during computation. The generation limit was set to ten times independent variable count and was also constrained by fifty stall generations. Thus, the computation reaches its end if there is no change to best fitness in fifty generations.

2.1. Polynomial equation

In this model a polynomial equation in the following format was used to classify loan applicants (Thomas et al., 2002):

$$f(x) = a_1 x_1^{b1} + a_2 x_2^{b2} + \cdots + a_D x_D^{bD} + c \tag{4}$$

The chromosome was formed by parameters we wished to estimate: $a_1, a_{2,} \ldots, a_D, b_1, b_2, \ldots, b_D$ and c. The initial ranges for a and b were set from -10 to 10 and from -100 to 100 for c. Once estimated the scoring function $f(x)$ was calculated for each chromosome and its value compared to a critical value which was set to 0. In the case that $f(x) > 0$, the applicant was classified as good. If $f(x) <= 0$, the applicant was marked as bad. The fitness score Φ was calculated based on Eq. (2). Given the vector D of 11 independent variables, each chromosome consisted of 23 genes in total.

2.2. Parameter range estimation

As defined by Yobas et al. (2000) in range estimation fitness function each chromosome consisted of adjacent blocks of genes. Each of these blocks related to a particular applicant characteristic, such as age, years at current employment, or nationality. In the case of a binary variable, a block consisted of two genes, a single value and an outcome value. For all other variables each block was formed by three genes: outcome value, minimum and maximum value respectively. The outcome variable was a binary value coded to 0 for 'do not include the characteristic to imply an outcome' or to 1 for 'use the range conditions specified by this characteristic to classify any applicant that meets them as *good*'. An example of a chromosome is shown in Table 1:

This chromosome is a simplified version of the actual one, consisting only of three characteristics as opposed to eleven. It indicates that characteristic age and years at current employment are to be used, whereas nationality will not be used to predict if the applicant belongs to a group of good payers. The chromosome reads as follows: an applicant is classified as *good* if he is aged

Table 1
Example of chromosome used by range estimation method.

Age	Years at current employment	Nationality
1 25 38	1 5 7	0 1

between 25 and 28, and has been working with his or her current employer from 5 to 7 years. Nationality is a binary variable representing either Venezuelan nationality or other, but in the case of the above, chromosome does not influence classification. A chromosome represents a solution in the form of a joint condition that has to be met in order to be classified as *good*; if any of the conditions are failed the case is classified as *bad*. The fitness score Φ was calculated from Eq. (2). A chromosome consisted of l genes, where $l = 2b + 3c$ where b is the number of binary characteristics, and c is the number of other than binary characteristics. Given the vector D of 11 independent variables, out of which three were binary, each chromosome consisted of 30 genes in total.

2.3. Weighted bitmask

Just as in the range estimation method, a chromosome in the bitmask model consists of adjacent blocks of genes that relate to a particular characteristic. Independently of the variable type, each characteristic is always represented by three genes. The first represents a bonus weight a, while the second represents a penalty weight b to be used in fitness calculation. The third gene represents the bitmask c of the possible values of the respective characteristic.

As defined previously, each customer x is characterized by D variables $x = (x_1, x_2, \ldots, x_D)$. Each characteristic x_j can acquire the value of exactly one element from a set V_j of all possible values for a given characteristic. Both the range estimation and the bitmask have as their goal the separation of the set V_j into two disjoint subsets: V_{Gj} and V_{Bj}, where V_{Gj} represents the subset of desirable characteristic values while V_{Bj} represents the subset of non-desirable characteristic values for a good client to have. This means that:

$$V_j = V_{Gj} \cup V_{Bj} \tag{5}$$

and

$$V_{Gj} \cap V_{Bj} = \emptyset \tag{6}$$

The bitmask c_j is then the binary representation of set V_j and subsets V_{Gj} and V_{Bj} in the following sense: its length in bits is equal to the number of elements n in set V_j and each bit represents exactly one of these elements. In the case that the element is part of subset V_{Gj} the respective bit's value is set to one; if the element is part of subset V_{Bj} its respective bit's value is set to zero.In the case of residential status there were three possible values: property owner, renting a property, or living with family. These were coded in the dataset as 4, 2 and 1 respectively using a binary system as shown in Table 2.

The bitmask then represents any of the above values or any combination of them with the exception of 0 as shown in Table 3.

This means that given a characteristic variable with n possible values the bitmask had to be constructed to represent $2^n - 1$ values, comprising of exactly n bits. Group of bits set to 1 then represent a positive condition.

Table 2
Variable coding.

Residential status	
100 – 4	Owner
010 – 2	Rented
001 – 1	Family

Table 3
Bitmask coding.

Residential status	
0 0 1	1 0 1
0 1 0	1 1 0
1 0 0	1 1 1
0 1 1	

The weight system was included to introduce a sense of different importance between elements from set V_j for any characteristic x_j.

In our case, the weights $a, b \in \langle 0, 10 \rangle$, but we treat a as a bonus and b as a penalty. The scoring function $f(x)$ based on a separation of variable ranges into good and bad parts, i.e. induced by the corresponding bitmask, is defined as follows:

$$f(x) = \sum_{j=1}^{D} q_j \tag{7}$$

where

$$q_j = a_j, \text{ if } x_j \in V_{Gj} \tag{8}$$

$$q_j = -b_j, \text{ if } x_j \in V_{Bj} \tag{9}$$

Once estimated the scoring function $f(x)$ was calculated for each chromosome, and its value compared to a critical value which was set to 0. In the case that $f(x) > 0$, the applicant was classified as good. If $f(x) <= 0$, the applicant was marked as bad. The fitness score Φ was calculated based on Eq. (2).

Putting the bitmask and weight system together, we can look at an example of a simplified chromosome shown in Table 4.

This chromosome indicates that characteristic 'residential status' has a bonus weight $a = 4$ and a penalty weight $b = 3$. The condition c to be met is 4 or a binary string (1 0 0), which means if the applicant lives in his own property, 4 is to be added to the score $f(x)$; in any other case 3 is to be deducted from $f(x)$. Similarly, characteristic 'years at current employment' has a bonus weight $a = 3$ and penalty weight $b = 6$. The condition c is represented by 224 or binary string (1 1 1 0 0 0 0 0), which implies that 3 is to be added to score $f(x)$ if the applicant works with his current company for any of the three highest possible periods (in our case 5, 6, or 7+ years), or in any other case 6 is to be subtracted. In the case where $a = b = 0$, as can be observed in characteristic 'owns a car', the variable is effectively excluded from the resulting model.

Given the vector D of 11 independent variables, each chromosome consisted of 33 genes in total.

The advantage of this type of encoding over the interval estimation is that the bitmask can address any possible subset of set V_j as opposed to the range estimation, which can only address subsets of V_j based on continuous intervals. Additionally, the weight system makes for more flexible and precise classification as in practice not all parameters are of equal importance. Thus, the ability of an applicant to meet certain criteria might be considered as being of more, less or equal importance.

Table 4
Example of chromosome used by bitmask method.

Residential status	Years at current employment	Owns a car
4 3 4	3 6 224	0 0 1

2.4. Data preparation

A sample of 489 credit card customers of a bank from Venezuela was used. The sample consisted of 344 clients classified as good and 145 cases classified as bad. A client was classified as bad if it had been in default for 90 days or more. All clients were characterized by 11 attributes summarized in Table 5.

It is worth noting that in some cases where data availability or quality is a problem, credit card data could be used as a substitute (Yoon & Kwon, 2010).

Using a sample of clients with known credit history requires that the customers involved in the study have been previously granted a credit card. Such a sample will contain only clients that have passed the current credit assessment criteria and thus previously rejected applicants will not be included. This may suggest a significant bias (Greene, 1998; Marshall, Tang, & Milne, 2010), but other research indicates such bias may be overestimated (Crook, Edelman, & Thomas, 2007; Reichert, Cho, & Wagner, 1983).

A cross validation technique was used to divide the data into ten stratified pairs of train and test samples. Each test sample contained 50 cases out of which 15 were bad and 35 good. The remaining 439 cases were used to train the genetic algorithm.

The genetic algorithm was trained ten times on each train sample. The best fitness chromosomes were recorded and subsequently tested on the corresponding test sample; the results were averaged.

The above procedure was repeated ten times to obtain ten observation results to be consolidated into one result per fitness function.

The data was encoded in two different ways. For interval estimation and polynomial equation, numerical variables were left as they were or scaled down by a constant. Alphabetical variables were replaced by numbers.

For the bitmask method, all data were encoded in binary system, where each possible value was represented by one bit position. Continuous variables were divided into eight intervals. All other variables had either two or three different possible values. The maximum value of the bitmask for each variable was then $2^n - 1$, where n was the number of bits (or number of possible values).

2.5. Evaluation criteria

Selecting the right evaluation criteria for model comparison is crucial when conducting research. In the case of credit scoring, various criteria can be used. One of the most popular is model accuracy (Marques et al., 2013). It is an easy to understand criterion, but it may not be necessarily the best for a practitioner attempting to construct a complete scoring card. For such a purpose the usage of the Gini coefficient, ROC curves, or the KS statistic may be more desirable (Crook et al., 2007). In the case of this paper, besides accuracy, model sensitivity and specificity were investigated as well. The construction and comparison of ROC curves was not possible as the range estimation method does not provide a score as its output, but only a binary value, which means the ROC curve would be reduced to a single point.

Table 5
Charcteristics variables.

Age
Marital status
Nationality
Education
Employment level
Years at current employment
Applicant's spouse working
Applicant's salary
Reference form other credit card
Residential status
Applicant owns a car

Table 6
Confusion matrix.

Predicted class	Observed class		Total
	Good	Bad	
Good	n_{GG}	n_{GB}	$n_{GG} + n_{GB}$
Bad	n_{BG}	n_{BB}	$n_{BG} + n_{BB}$
Total	$n_{GG} + n_{BG}$	$n_{GB} + n_{BB}$	

Table 7
Accuracy in %.

Obsv.	Polynom	Range	Bitmask
1	74.18	69.58	76.10
2	74.42	69.60	76.04
3	73.92	69.54	75.94
4	73.96	69.44	75.98
5	73.52	69.40	75.46
6	73.92	69.54	76.28
7	74.14	69.38	74.82
8	74.38	69.50	76.16
9	73.98	69.30	76.12
10	74.60	69.52	75.58
Mean	74.10	69.48	75.85

Based on the confusion matrix in Table 6, accuracy, sensitivity and specificity were defined as follows. Accuracy is represented by correctly classified good and bad from total population:

$$Ac = (n_{GG} + n_{BB})/(n_{GG} + n_{BB} + n_{GB} + n_{BG}) \quad (10)$$

Specificity is the percentage of correctly classified good clients:

$$Sp = n_{GG}/(n_{GG} + n_{BG}) \quad (11)$$

Sensitivity is represented by the percentage of correctly classified bad cases:

$$Se = n_{BB}/(n_{BB} + n_{GB}) \quad (12)$$

In the field of credit scoring, it is particularly important to investigate method performance with respect to classification power in terms of good and bad customers separately as the misclassification costs can be significantly different depending on the type of error committed.

2.6. Statistical methods

Model accuracy was measured for each observation. As stated in Section 2.4, each observation was comprised of measurements on ten test samples. In total we produced ten independent observations for each fitness function model. The paired t-Test and the Wilcoxon matched-pairs sign rank test were selected to assess the statistical significance of differences in model accuracy. In addition, the Shapiro–Wilk normality test was employed to test if the samples followed a Gaussian distribution. An overview of learning algorithms evaluation and the description of selected statistical significance tests can be found in Japkowicz and Shah (2011).

3. Results

All computations were performed in MATLAB®[1] using custom built evolution, selection, creation, crossover, mutation, and fitness

[1] MATLAB® is a registered trademark of The MathWorks, Inc.

functions. Results for the main criterion accuracy are summarized in Table 7.

From Table 7 we can see that the best accuracy was achieved by the bitmask model at 75.85%, while the polynomial equation was 74.1% accurate and the last was the interval estimation which was only 69.48% accurate. These results were validated by conducting the paired t-Test and the resulting statistics values for each pairwise comparison can be found in Table 8.

The null hypothesis is that there is no difference between the models. Given the tabulated critical value of 3.2498 for the two tailed t-Test at significance level $\alpha = 0.01$, and the fact that all the computed statistics in Table 8 are greater, we can reject the null hypothesis. This test, however, assumes a normal distribution among the data. To test the normality assumption we used the Shapiro–Wilk normality test: the resulting p-values are summarized in Table 9.

We can see that the normality assumption holds for all models at significance level $\alpha = 0.01$ but at significance level $\alpha = 0.05$ it has to be rejected for the bitmask model. Due to this we have also conducted a non-parametric Wilcoxon matched-pairs sign rank test; its corresponding p-values are recorded in Table 10.

Since all p-values are lower than 0.01 we can reject the null hypothesis that there are no disparities in accuracy among the different fitness function models at significance level $\alpha = 0.01$.

Since the cost of misclassifying a bad customer as good when conducting credit scoring is usually greater than misclassifying a good customer as bad (Thomas et al., 2002; Vojtek and Kocenda, 2006), it is especially important to be able to predict the group of bad customers accurately.

Table 11 shows the accuracy, specificity and sensitivity of the respective models calculated based on Eqs. (10)–(12) respectively. It can be observed that the bitmask model at our given cut-off point has both greater sensitivity and specificity than the range estimation model. The bitmask model also has better sensitivity than the polynomial model: when compared with the latter, the former displays 9.96% greater sensitivity at the cost of only 1.77% in specificity.

The final characteristic ability of a given model to train itself and adapt to a different data type is recorded in Table 12. From the table it is evident that when it came to the ability to train, just as in the accuracy over test sample, the bitmask model ranked first, with 81.64% correctly classified cases. It was followed by the polynomial model with 78.81% and finally by range estimation with 71.80%.

Table 8
Paired t-Test statistics value.

	Range	Bitmask
Polynom	51.7489	10.4961
Range	x	48.7169

Table 9
Shapiro Wilk normality test.

	Polynom	Range	Bitmask
p-value	0.7896	0.5530	0.0242

Table 10
Wilcoxon signed rank p-value.

	Range	Bitmask
Polynom	0.0020	0.0020
Range	x	0.0020

Table 11
Accuracy, sensitivity and specificity in %.

	Accuracy	Specificity	Sensitivity
Bitmask	75.85	85.28	53.85
Polynom	74.10	87.05	43.89
Range	69.48	79.65	45.75

Table 12
Ability to train in %.

Obsv.	Polynom	Range	Bitmask
1	78.81	71.81	81.61
2	78.75	71.74	81.62
3	78.82	71.84	81.60
4	78.84	71.78	81.61
5	78.78	71.77	81.64
6	78.84	71.80	81.56
7	78.82	71.81	81.60
8	78.85	71.83	81.80
9	78.73	71.82	81.70
10	78.87	71.81	81.69
Mean	78.81	71.80	81.64

4. Discussion and conclusions

Recent research reviews of modeling techniques in the area of credit scoring indicate an increase in the popularity of hybrid techniques (Marques et al., 2013; Sun et al., 2014). Genetic algorithms in this context often take on a supporting role for other techniques as neural networks. This trend is the result of the fact that more promising results have been reported using other methods or their combination. However, most previously published research about the use of genetic algorithms for credit scoring does not give enough detail regarding the fitness function used in their optimization.

One of the main contributions of this paper is the fact that it aims at partially filling the gap by concentrating on fitness function performance evaluation. Since fitness function and data encoding are crucial for genetic algorithm performance and should always reflect the problem at hand, its proper modification can significantly improve the GAs results as a standalone method. The paper proposes a modified fitness function model based on a variable bitmask, investigates its performance, and compares it with two approaches that have been previously published. The results obtained indicate that the bitmask model in accuracy outperforms both the interval and polynomial model. The performed paired t-Test and Wilcoxon matched-pairs sign test indicated statistically significant differences between methods. Together with better sensitivity, the bitmask model comes out as a promising alternative fitness function method. In practical usage this can have significant impact given the high costs associated with misclassifying a credit customer as each percent advantage may be converted into considerable profit. This is especially true when we are able to identify the group of potential bad payers more accurately, as research and practice indicate the cost of classifying a bad client as good is significantly higher than committing the opposite error.

The main research limitations when it comes to credit scoring are database availability as well as quality and computational requirements. In the case of the bitmask model, computational requirements are of concern; these depend directly on the database's characteristic variables and their maximum possible values n. Because the maximum size of the bitmask is 2^n, the complexity of the optimization problem rises exponentially. For larger datasets with complex discrete variables (such as home address), significantly stronger computing power may be needed. It is also more difficult to group this type of variables into reasonable intervals and the need to recode the dataset is one of the main limitations of the proposed model.

The bitmask method is based on a division of a set of characteristics into two subsets and a subsequent weight system application. Additional research in this topic could be beneficial as more subsets each with its own weight might possibly lead to further performance improvement. In context with the previously mentioned limitations, different subset representation might be investigated as well as different data encoding methodology as it could significantly ease the computational requirements.

Acknowledgements

Computational resources were provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144. The author gratefully thanks the anonymous referees for their useful comments and the editors for their work.

References

Avery, R. B., Calem, P. S., & Canner, G. B. (2004). Consumer credit scoring: Do situational circumstances matter? *Journal of Banking & Finance, 28*(4), 835–856. http://dx.doi.org/10.1016/j.jbankfin.2003.10.009.

Ben-David, A., & Frank, E. (2009). Accuracy of machine learning models versus "hand crafted" expert systems – A credit scoring case study. *Expert Systems with Applications, 36*(3), 5264–5271. http://dx.doi.org/10.1016/j.eswa.2008.06.071.

Blöchlinger, A., & Leippold, M. (2006). Economic benefit of powerful credit scoring. *Journal of Banking & Finance, 30*(3), 851–873. http://dx.doi.org/10.1016/j.jbankfin.2005.07.014.

Bonfim, D. (2009). Credit risk drivers: Evaluating the contribution of firm level information and of macroeconomic dynamics. *Journal of Banking & Finance, 33*(2), 281–299. http://dx.doi.org/10.1016/j.jbankfin.2008.08.006.

Chen, M.-C., & Huang, S.-H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications, 24*(4), 433–441. http://dx.doi.org/10.1016/S0957-4174(02)00191-4.

Chi, B.-W., & Hsu, C.-C. (2012). A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model. *Expert Systems with Applications, 39*(3), 2650–2661. http://dx.doi.org/10.1016/j.eswa.2011.08.120.

Chuang, C.-L., & Lin, R.-H. (2009). Constructing a reassigning credit scoring model. *Expert Systems with Applications, 36*(2), 1685–1694. http://dx.doi.org/10.1016/j.eswa.2007.11.067.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*(3), 1447–1465. http://dx.doi.org/10.1016/j.ejor.2006.09.100.

Desai, V. S., Conway, D. G., Crook, J. N., & Overstreet, G. A. (1997). Credit-scoring models in the credit-union environment using neural networks and genetic algorithms. *IMA Journal of Management Mathematics, 8*, 323–346. http://dx.doi.org/10.1093/imaman/8.4.323.

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37. http://dx.doi.org/10.1016/0377-2217(95)00246-4.

Dinh, T. H. T., & Kleimeier, S. (2007). A credit scoring model for Vietnam's retail banking market. *International Review of Financial Analysis, 16*(5), 471–495. http://dx.doi.org/10.1016/j.irfa.2007.06.001.

Dionne, G., Artís, M., & Guillén, M. (1996). Count data models for a credit scoring system. *Journal of Empirical Finance, 3*(3), 303–325. http://dx.doi.org/10.1016/0927-5398(96)00004-7.

Finlay, S. (2009). Are we modelling the right thing? The impact of incorrect problem specification in credit scoring. *Expert Systems with Applications, 36*(5), 9065–9071. http://dx.doi.org/10.1016/j.eswa.2008.12.016.

Finlay, S. (2010). Credit scoring for profitability objectives. *European Journal of Operational Research, 202*(2), 528–537. http://dx.doi.org/10.1016/j.ejor.2009.05.025.

Fogarty, T. C., & Ireson, N. S. (1993). Evolving Bayesian classifiers for credit control—A comparison with other machine-learning methods. *IMA Journal of Management Mathematics, 5*(1), 63–75. http://dx.doi.org/10.1093/imaman/5.1.63.

Gordini, N. (2014). A genetic algorithm approach for SMEs bankruptcy prediction: Empirical evidence from Italy. *Expert Systems with Applications, 41*(14), 6433–6445. http://dx.doi.org/10.1016/j.eswa.2014.04.026.

Greene, W. (1998). Sample selection in credit-scoring models. *Japan and the World Economy, 10*(3), 299–316. http://dx.doi.org/10.1016/S0922-1425(98)00030-9.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press.

Jacobson, T., & Roszbach, K. (2003). Bank lending policy, credit scoring and value-at-risk. *Journal of Banking & Finance, 27*(4), 615–633. http://dx.doi.org/10.1016/S0378-4266(01)00254-0.

Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms: A classification perspective*. Cambridge University Press.

Kim, Y. S., & Sohn, S. Y. (2004). Managing loan customers using misclassification patterns of credit scoring model. *Expert Systems with Applications, 26*(4), 567–573. http://dx.doi.org/10.1016/j.eswa.2003.10.013.

Lopez, J. A., & Saidenberg, M. R. (2000). Evaluating credit risk models. *Journal of Banking & Finance, 24*(1–2), 151–165. http://dx.doi.org/10.1016/S0378-4266(99)00055-2.

Marques, A. I., Garcia, V., & Sanchez, J. S. (2013). A literature review on the application of evolutionary computing to credit scoring. *Journal of the Operational Research Society, 64*(9), 1384–1399. http://dx.doi.org/10.1057/jors.2012.145.

Marshall, A., Tang, L., & Milne, A. (2010). Variable reduction, sample selection bias and bank retail credit scoring. *Journal of Empirical Finance, 17*(3), 501–512. http://dx.doi.org/10.1016/j.jempfin.2009.12.003.

Michalewicz, Z. (1996). *Genetic algorithms + data structures = Evolution programs*. Springer Science & Business Media (pp. 387). .

Mitchell, M. (1998). *An introduction to genetic algorithms*. Cambridge: MIT Press.

Oreski, S., & Oreski, G. (2014). Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert Systems with Applications, 41*(4), 2052–2064. http://dx.doi.org/10.1016/j.eswa.2013.09.004.

Oreski, S., Oreski, D., & Oreski, G. (2012). Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. *Expert Systems with Applications, 39*(16), 12605–12617. http://dx.doi.org/10.1016/j.eswa.2012.05.023.

Reichert, A. K., Cho, C.-C., & Wagner, G. M. (1983). An examination of the conceptual issues involved in developing credit-scoring models. *Journal of Business & Economic Statistics, 1*(2), 101–114. http://dx.doi.org/10.1080/07350015.1983.10509329.

Stiglitz, J., & Weiss, A. (1981). Credit rationing in markets with imperfect information. *American Economic Review, 71*(3), 393–410.

Sun, J., Li, H., Huang, Q.-H., & He, K.-Y. (2014). Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches. *Knowledge-Based Systems, 57*, 41–56. http://dx.doi.org/10.1016/j.knosys.2013.12.006.

Sustersic, M., Mramor, D., & Zupan, J. (2009). Consumer credit scoring models with limited data. *Expert Systems with Applications, 36*(3), 4736–4744. http://dx.doi.org/10.1016/j.eswa.2008.06.016.

Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). *Credit scoring and its applications*. Philadelphia: SIAM.

Vojtek, M., & Kocenda, E. (2006). Credit scoring methods. *Czech Journal of Economics and Finance, 56*(3–4), 152–167.

Vukovic, S., Delibasic, B., Uzelac, A., & Suknovic, M. (2012). A case-based reasoning model that uses preference theory functions for credit scoring. *Expert Systems with Applications, 39*(9), 8389–8395. http://dx.doi.org/10.1016/j.eswa.2012.01.181.

Wu, D., & Olson, D. L. (2010). Enterprise risk management: Coping with model risk in a large bank. *Journal of the Operational Research Society, 61*(2), 179–190. http://dx.doi.org/10.1057/jors.2008.144.

Yobas, M. B., Crook, J. N., & Ross, P. (2000). Credit scoring using neural and evolutionary techniques. *IMA Journal of Management Mathematics, 11*(2), 111–125. http://dx.doi.org/10.1093/imaman/11.2.111.

Yoon, J. S., & Kwon, Y. S. (2010). A practical approach to bankruptcy prediction for small businesses: Substituting the unavailable financial data for credit card sales information. *Expert Systems with Applications, 37*(5), 3624–3629. http://dx.doi.org/10.1016/j.eswa.2009.10.029.