Expert Systems with Applications 39 (2012) 8389–8395

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

A case-based reasoning model that uses preference theory functions for credit scoring

Sanja Vukovic *, Boris Delibasic, Ana Uzelac, Milija Suknovic

University of Belgrade, Faculty of Organizational Sciences, Jove Ilica 154, Belgrade, Serbia

ARTICLE INFO

Keywords:
Case-based reasoning
Preference functions
Genetic algorithm
Credit scoring
Classification

ABSTRACT

We propose a case-based reasoning (CBR) model that uses preference theory functions for similarity measurements between cases. As it is hard to select the right preference function for every feature and set the appropriate parameters, a genetic algorithm is used for choosing the right preference functions, or more precisely, for setting the parameters of each preference function, as to set attribute weights. The proposed model is compared to the well-known k-nearest neighbour (k-NN) model based on the Euclidean distance measure. It has been evaluated on three different benchmark datasets, while its accuracy has been measured with 10-fold cross-validation test. The experimental results show that the proposed approach can, in some cases, outperform the traditional k-NN classifier.

1. Introduction

The number of risks banks face every day is constantly growing. Banks are financial organizations which turn risk into profits. Most of their revenues are generated by credits, i.e. lending operations. Credit lending is therefore one of the most important profit generators for a bank. Naturally, the main risk when granting a loan is that the clients will not be able to fulfil their obligations towards the bank and that the bank will lose its funds.

During the last couple of decades, a rapid growth has been noticed in both availability and use of credits. Until recently, the decision to grant a credit was based on human judgement to assess the risk of default (Thomas, 2000). The growth in the demand for credit, however, has led to a larger interest in the use of more formal and objective methods (generally known as credit scoring) to help credit providers decide whether or not to grant credit to an applicant (Akhavein, Frame, & White, 2005; Chye, Chin, & Peng, 2004).

Credit scoring is a classification problem. That is why credit scoring models help to decide whether to grant a credit to new applicants, considering the customer's characteristics such as age, income and marital status (Chen & Huang, 2003). Credit granting is a very important part of a bank activity, as it may yield big profits, but there is also a significant risk involved in making decisions in this area and the mistakes may be very costly for financial institutions (Zakrzewska, 2007).

For all the above reasons, the decision-making related to credit granting is one of the crucial elements in the policy of each bank. The key problem is to distinguish between good (that surely repay) and bad (that likely default) credit applicants. This means that credit risk evaluation consists of building classification rules that properly define bank customers as good or bad payers (Zakrzewska, 2007).

For many years, the decision whether to grant a loan has been done by credit analysts. The analysts usually had to write down the rules they used to assess a loan applicant's credibility in repaying the loan. Credit decisions were made using these rules.

Credit scoring methodology can be used for different purposes, such as credit cards, personal loan applications, home loans, small business loans, as well as insurance applications and renewals. Furthermore, it can be used to increase the response rate to advertising campaigns, etc. (Thomas, 2000). Therefore, it is essential to find a way to build an effective customer classification model that can predict the customer's behaviour more accurately.

There are a number of models which can be used for credit evaluation in the banking industry. Some of these methods are statistical, while some of them rely on artificial intelligence (AI) approaches. The statistical methods often used for credit scoring are multiple regression (e.g. Meyer & Pifer, 1970), discriminant analysis (e.g. Altman, 1968; Banasik, Crook, & Thomas, 2003), and logistic regression (e.g. Desai, Crook, & Overstreet, 1996; Dimitras, Zanakis, & Zopounidis, 1996; Elliott & Filinkov, 2008; Lee, Chiu, Lu, & Chen, 2002; Martin, 1977), while the AI methods include inductive learning (e.g. Han, Chandler, & Liang, 1996; Shaw & Gentry, 1998), artificial neural networks (e.g. Boritz & Kennedy, 1995; Coakley & Brown, 2000; Jo & Han, 1996; Lee & Chen, 2005; West, 2000; Zhang, Hu, Patuwo, & Indro, 1999), genetic algorithms (e.g.

* Corresponding author.
E-mail addresses: sanja.vukovic@gmail.com (S. Vukovic), boris.delibasic@fon.bg.ac.rs (B. Delibasic), ana.uzelac@sf.bg.ac.rs (A. Uzelac), milija.suknovic@fon.bg.ac.rs (M. Suknovic).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.181

Desai, Convay, Crook, & Overstreet, 1997; Huang, Chen, & Wang, 2007; Huang, Tzeng, & Ong, 2006; Yobas, Crook, & Ross, 2000), and artificial immune system (e.g. Leung, Cheong, & Cheong, 2007).

Since the seminal work of Schank and Abelson (1977), case-based reasoning (CBR) has been successfully applied in many areas including credit scoring (e.g. Bryant, 1997; Buta, 1994; Shin & Han, 2001; Wheeler & Aitken, 2000).

CBR is one of the methods which can be successfully applied to financial problems such as credit scoring. It can also be used in many other areas, such as customer segmentation (e.g. Changchien & Lin, 2005; Chiu, 2002; Chun & Park, 2006), medical and manufacturing industry (e.g. Hsu, Chiu, & Hsu, 2004; Im & Park, 2007; Tseng, Chang, & Chang, 2005), etc.

Despite its many advantages, there are some problems that must be solved in order to design an effective CBR system (Ahn & Kim, 2008):

- How to select the appropriate similarity function to generate classification from stored cases?
- How to select the appropriate features, known as feature selection?
- How to determine the weight of each feature, which is known as feature weighting?
- How to determine the optimal k parameter if k-nearest neighbour (k-NN) algorithm is used?
- How to compute similarity for categorical variables which could, among the numerical variables, also describe cases?

There have been many studies attempting to resolve these problems. The selection of the appropriate similarity measures, and the choice of feature subsets and their weights in the case of the retrieval step have been the most popular research issues (e.g. Ahn, Kim, & Han, 2007; Chiu, Chang, & Chiu, 2003; Kim & Han, 2001; Liao, Zhang, & Mount, 1998; Shin & Han, 1999; Wang & Ishii, 1997).

Similarity measurement is an important part of every CBR model. Therefore, it is usually used the Euclidean metric. Preference theory can also be used for measuring similarity between cases, especially as it provides more opportunities in expressing the decision-makers' preferences. Li, Sun, and Sun (2009) and Li and Sun (2010) proposed combining CBR with preference functions (outranking relations) for financial distress and business failure prediction, respectively.

Li et al. (2009) used the outranking relations based on the preference function in Electre III, while Li and Sun (2010) constructed a hybrid CBR forecasting system which used all the available preference functions in outranking approaches, such as Electre, Promethee, and Oreste. In this paper, we have used the preference functions proposed in method Promethee for measuring similarity between cases.

Li and Sun (2010) used a trial-and-error iterative process to identify the optimal hybrid CBR module with corresponding preference function and parameters. In this paper, genetic algorithm has been used for such purposes.

It has been interesting to consider whether the domain knowledge, which can be expressed through preference functions, could be better exploited in such a way to improve the predictive performance of a CBR system.

The main difference between CBR with preference functions and the traditional CBR (k-NN) is the mechanism of similarity computation. We hypothesize that the use of preference theory functions in CBR can show better results than the traditional k-NN, based on the Euclidean distance measure in loan granting.

We have also analysed the number of neighbours that are taken into account for classification, as well as the influence of attribute (feature) weights on the accuracy.

The remainder of this paper is organized as follows: Section 2 introduces the basic concepts of the methods used in this paper. Section 3 proposes a new hybrid method that combines CBR and preference functions, with support of GA. Experimental evaluation of the model on benchmark credit scoring data is also presented in this section. Finally, some concluding remarks and the ideas for future research are discussed in Section 4.

2. Basic concepts

In this study, we evaluate the usefulness of CBR model with combining preference functions and a genetic algorithm (GA). The first two parts of this section present a review of basic concepts of CBR, with the emphasis on case retrieval, as this is the main topic of consideration in this investigation. The third part describes the main types of preference functions. The final, fourth part, describes the GA setup that has been used in this paper.

2.1. Case-based reasoning

Credit scoring methodology requires experience-based expertise. When solving a new problem, the experts rely on the past scenarios. They need to know which credits have been successful and which have failed. They also need to know how to modify an old case to fit the new situation. CBR is a general paradigm for experience-based reasoning. It assumes a memory model for representing, indexing, and organizing the past cases and a process model for retrieving and modifying the old cases and assimilating the new ones (Slade, 1991).

CBR solves new problems by relating some previously solved problems or experiences to the new problems thus forming analogical inferences for problem solving (Kolodner & Mark, 1992). Facing a new problem, CBR retrieves similar cases stored in a case base and adapts them to the new problem. The key factors affecting the performance of a CBR retrieval mechanism are case representation, case indexing and similarity metric (Buta, 1994).

CBR is generally quite simple to implement and can often handle complex and unstructured decisions very effectively (Ahn et al., 2007).

The retrieval of relevant previous cases is crucial to the success of a CBR system. The aim of case-based retrieval is to retrieve the most useful previous cases towards the optimal resolution of a new case and to ignore those previous cases that are irrelevant (Montazemi & Gupta, 1997).

A good retrieving function should take into account the features of a case that are more important. The case that matches the important features but not the less important ones will almost certainly be a better match than the one that matches less important features but does not match the important ones. For this reason, the integration of domain knowledge into the case matching and retrieving function is highly recommended in modelling a successful CBR system (Park & Han, 2002).

Solving a problem by CBR involves obtaining a problem description, measuring the similarity of the current problem to the previous problems stored in a case base (or memory) with their known solutions, retrieving one or more similar cases and attempting to reuse the solution of one of the retrieved cases, possibly after adapting it to account for differences in problem descriptions. The solution proposed by the system is then evaluated (e.g., by being applied to the initial problem or assessed by a domain expert). Then, if the proposed solution is adequate the problem description and its solution can be retained as a new case, and the system has learned to solve a new problem (Lopez de Mantaras et al., 2005).

According to Kolodner (1993), CBR comprises four major steps: (1) case representation, (2) case indexing, (3) case retrieval, and (4) case adaptation. Case representation represents the features associated with a past case; case indexing intends to facilitate the search and retrieval of similar cases; case retrieval retrieves the cases most similar to the studied case from the database; and case adaptation is a process of modifying an existing case or building a new one if all the retrieved cases do not comply with the case encountered (Chen, Wang, & Feng, 2010).

Aamodt and Plaza (1994) suggested a model of the problem solving cycle in CBR, which involved four tasks, known as 4 REs (i.e. retrieve the most similar case or cases, reuse the information and knowledge from that case to solve the problem, revise the proposed solution and retain the parts of this experience likely to be useful for future problem solving). Reinartz, Iglezakis, and Roth-Berghofer (2001) revised this model and extended it by including two new steps – a review step in order to monitor the quality of system knowledge and a restore step, which selects and applies maintenance operations.

One of the most important steps in the CBR cycle, which will be the main focus of this work, is the retrieval of previous cases that can be used to solve the target problem. The basic assumption on which the similarity-based retrieval rests is that the most similar cases are the most useful ones for solving the target problem.

2.2. Case retrieval by using k-NN

One of the most useful capabilities of CBR is its ability to efficiently retrieve relevant cases from the case library. Successful performance of the retrieval mechanism depends on good representation, indexing and similarity metric (Park & Han, 2002). Three methods are usually used for case retrieval: nearest neighbour (NN), inductive learning, and knowledge-guide, or the combination of these (Buta, 1994). We use the NN method for case retrieval as it is the most commonly used method.

NN can be used for classifying a new case by searching for the cases in the database that have the most similar features and assigning the class of its nearest neighbour(s) to the new case.

Each case from the database has a classification (categorical) variable of interest (e.g. creditworthy or not), and a number of additional predictor variables (e.g. age, income, marital status, etc.). For a new case, which is to be classified, NN algorithm locates the closest member from the database and assigns the category of its nearest neighbour to the new case.

An important improvement in classification accuracy can be achieved with feature weighting algorithms. This means that the most relevant features have the highest weights. The overall similarity determined by a weighed NN matching function is mathematically represented as follows (Kolodner, 1993):

$$\text{Similarity }(T,S) = \sqrt{\sum_{i=1}^{F} w_i (T_i - S_i)^2} \tag{1}$$

where w_i is the weight of feature (attribute) i, T is the target (input) case, S is the source (retrieved) case, F is the number of attributes in each case, and i is an individual feature from 1 to F.

Setting the weights in the similarity function appropriately can improve the performance of the NN algorithms. More important attributes should be assigned larger weights than the less important ones, while the totally irrelevant attributes should be assigned zero weights (Park & Han, 2002).

k-NN method is a modified, extended version of NN that gives better classification results because it searches for k closest cases to the new case (k is usually an odd number) and proposes the solution that occurs most frequently in the k cases (Chen et al., 2010). For instance, in the credit scoring problem, where we should make a decision whether to grant a loan or not, k-NN method is adopted to infer whether our decision should be approved or rejected. In other words, after a new problem (case) shows up, the CBR system (on the basis of similarity function) will retrieve k most similar cases from the database (k-NN) and infer to which class the new case belongs. The group to which the majority of the k cases belong is the inferred group to which the new case belongs (Chen et al., 2010).

k-NN method uses a similarity function to generate classification from the stored cases. Several studies have shown that k-NN performance is highly sensitive to the choice of this function, so in order to reduce this sensitivity many k-NN methods have been proposed by using various distance functions.

2.3. Preference functions

The main weakness of the traditional (a.k.a. pure) CBR is the process of validation based on the Euclidean norm, where each criterion is measured equally. The only way to show certain preference to a certain criterion was by feature weighting. However, that has often been insufficiently reliable for solving practical problems.

Accordingly, there is a kind of over-generalization in distance computation, which means that the expert's indifference or strict preference towards differences between two cases cannot be properly expressed. On the other hand, validity constraints are the conditions (i.e., domain knowledge) that prevent over-generalization. Validity constraints either specialize the previous case or limit its applicability (Montazemi & Gupta, 1997).

Domain knowledge can be incorporated in constructing the CBR model, either by the relative importance of criteria or by outranking relations (OR), including the strict difference, weak difference, and indifference, between the cases on each feature.The parameters for OR are acquired by the domain experts, or they can be determined, for example, by a GAs approach, as proposed in this paper.

It is difficult to determine the experts' validity constraints in an ill-structured decision-making problem. The problem is in the choice of a validity constraint and its parameterization (threshold of indifference and/or threshold of strict preference) for each attribute.

This poses a considerable difficulty, because the expert must select boundaries from a large number of possible values. We therefore propose using GA for the objective choice and parameterization of validity constraints.

For measuring similarity between cases, we use the preference functions as validity constraints proposed in the famous multicriteria decision-making method – the Promethee (Brans & Vincke, 1985; Mareschal, 1986, 1988; Mareschal & Brans, 1988). For each attribute, a different preference function can be defined.

In Promethee preference functions are based on pair-wise comparisons of the alternatives along each recognized criterion (in credit scoring problem, there are comparisons of the feature's values between the target case and each historical case from the database). For each criterion, the preference function translates the difference between the values obtained by two alternatives into a preference degree ranging from zero to one.

For each feature and each existing case in the base, in the classical approach, the distance between a new case and its features and the existing cases and their features, is measured as follows:

$$d_i = T_i - S_i \tag{2}$$

where d_i is the difference between T – target (input) case and S – source (retrieved) case on the ith feature (attribute)The Promethee method extends the classical approach by allowing different preference functions $p_i(d_i)$ for each criterion, so each attribute reflects a preference level of S_i over T_i that ranges from 0 to 1. If

$p_i(d_i) = 0$, both alternatives are considered indifferent to each other on the ith attribute. If $p_i(d_i) = 1$, one alternative is strictly preferred to the other on the ith attribute. Some preference functions have an indifference interval, defined by a threshold q_i which allows to show indifference between two alternatives on an attribute not only when they have the same values considering the ith attribute (i.e., $d_i = 0$), but also when they are just similar to each other ($0 < d_i < q_i$). Furthermore, these functions can have a smooth transition between indifference and strict preference, which permits making preference judgments with distinct intensities. Values between 0 and 1 express the preference intensity in such a way that $p_i(d_i) \sim 0$ indicates weak preference and $p_i(d_i) \sim 1$ indicates strong preference (Parreiras & Vasconcelos, 2007).

Some authors (Brans, Vincke, & Mareschal, 1986) propose the main types of preference functions, which cover most of the practical situations (Fig. 1).

Zero, one or two parameters have to be defined for each preference function. The meaning of such parameters is

- q is a threshold of indifference;
- p is a threshold of strict preference and
- s is an intermediate value between p and q.

The q indifference threshold is the largest deviation, which is considered negligible, while the p preference threshold is the smallest deviation, which is considered sufficient to generate a full preference.

In this paper, the fifth type of preference function is chosen, as this function is the most important and it attracts the largest number of theoretical and practical applications for the evaluations carried out by the Promethee methods (Podvezko & Podviezko, 2010). The type 5 preference function uses the threshold of indifference and the threshold of strict preference that have to be defined. In this study it has been done by using GA.

2.4. Genetic algorithms

GA approach is the optimization method that includes a stochastic search technique that has the ability to search large and complicated spaces. It improves the search results by constantly trying various possible solutions with genetic operations (Ahn et al., 2007) similar to natural genetics and evolutionary principles (Ahn & Kim, 2008). GA basically explores a complex space in an adaptive way through GA operators: selection, crossover and mutation. This algorithm uses natural selection – survival of the best-fit individuals, in order to solve optimization problems (Kim, 2004). GA approach is particularly suitable for parameter optimization problems with an objective function subject to various hard and soft constraints (Shin & Han, 1999).

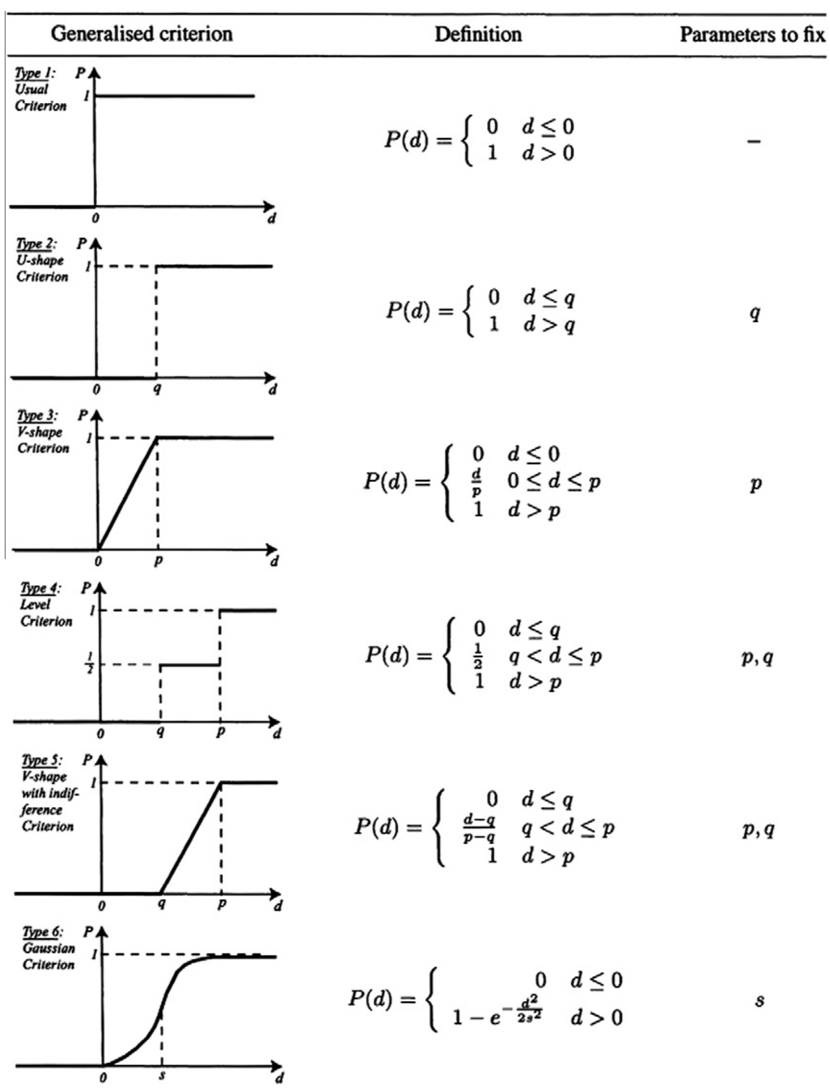

Fig. 1. Types of preference functions.

Generally, the process of GA develops in the following manner (Ahn et al., 2007):

First, a GA randomly generates a set of solutions which represent the initial population. Each solution in the population is called a chromosome and it is usually given in the form of a binary string. After generating the initial population, GA computes the fitness function of each chromosome. The fitness function is a user-defined function that returns the evaluation results of each chromosome, so that the higher fitness value means that its chromosome is better. Classification accuracy (performance) is typically used as the fitness function for classification problems.

After generating the initial population, the offspring is generated by the application of genetic operators. Among various genetic operators – selection, crossover and mutation are the most fundamental and most popular operators. The selection operator determines which chromosome will survive. In the crossover, substrings from pairs of chromosomes are exchanged thus producing new pairs of chromosomes. In the mutation, which usually has a small mutation rate, arbitrarily selected bits in a chromosome are inverted. These steps of evolution continue until the stopping conditions are satisfied. In most cases, the stopping criterion is set to the maximum number of generations (Chiu, 2002; Fu & Shen, 2004; Han & Kamber, 2001).

GA approach is used in this study to determine the appropriate parameters for preference functions, as to find the weights of features. Classification accuracy of the model is set as the fitness function for GA.

3. The research design and experiments

3.1. Combining CBR and preference theory functions with GA support

The proposed hybrid model can be abbreviated as CBR–PF–GA (combination of CBR and preference theory functions with the use of GA).

The model with equal importance (weight) of each feature is called (pure) CBR–PF–GA, while the model with weighted features is called weighted CBR–PF–GA. The latter approach allows us to have the information, not only about preferences within each criterion, but also about preferences among the criteria (weights of relative importance of different criteria, weight of feature).

The proposed model is developed by using Microsoft Excel 2003 and Palisade Software's Evolver Version 5.5. (Palisade Software, www.palisade.com)

The process of building a CBR–PF–GA classification model is as follows:

Step 1. Entering the basic information about the dataset.

Weights have to be defined for each attribute, as well as the type of preference function with their parameters (e.g. p, q from Fig. 1). To determine the right weights and preference functions' parameters, we have used the GA from Evolver.

Step 2. Entering the GA parameters (population size, the stopping criteria, etc.) and performing the CBR process.

For case retrieval, k-NN method is used. Since the focus of this paper is not on the choice of optimal k, we have tested the proposed model on several k values, varying from 1 to 9 on every odd value.

NN model is used for measuring the similarity between the cases. For each categorical feature, the overlap function is used (i.e. the distance is 1 if the compared values are different, and 0 if the compared values are the same). For numerical attributes, our model CBR–PF–GA uses preference function type 5.

For each case from the test set, the model should assign one of the two possible classes, by measuring the distance of that case from every case in the training set. For estimating the performance of a predictive model, 10-fold cross-validation (CV) methodology is used.

Step 3. Calculating the fitness value for each chromosome.

Step 4. Applying genetic operators and producing a new generation. In step 5, the process of GAs evolution goes on towards the direction of maximizing the value of the fitness function. It includes selection of the fittest, crossover and mutation. By applying these operators, a new generation of the population is produced (Ahn et al., 2007).

Step 5. Steps 3–5 are repeated until the stopping conditions have been are satisfied. Once the evolving process has been finished, according to the stopping criteria, the best parameters – the information about the parameters of the preference functions and the weights of features – are stored.

3.2. The datasets

We have used three datasets illustrated in Table 1. The Australian and German credit datasets are available from the UCI Repository of Machine Learning Databases (Frank & Asuncion, 2010), while SPSS dataset is from SPSS tutorial (SPSS for Windows 13.0, 2004).

The Australian credit data consist of 307 instances of creditworthy applicants and 383 samples where credit should not be extended. Each instance contains six nominal, eight numeric and one class attribute (accepted or rejected).

The German credit dataset contains 1000 cases, where 700 samples were accepted and maintained good credit, while 300 samples were accepted, but became delinquent. Each case is described with a set of 20 different attributes, which describe the credit history, account balances, loan purpose, loan amount, employment status, personal information, age, housing, and job title.

The SPSS credit dataset contains information on 700 customers who were previously given loans. It is less balanced, consists of 517 instances of creditworthy applicants and 183 instances where credit should not be granted.

3.3. Experimental results

To test the effectiveness of our model, four different CBR models have been applied:

1. Unweighted CBR (Pure CBR) where all features are weighted equally and the Euclidean distance function is used.
2. Unweighted CBR with preference functions optimized with GA (Pure CBR–PF–GA) where all the attributes have the same weight.
3. Weighted CBR with Euclidean distance function and weighted features (generated by using GA).
4. Weighted CBR–PF–GA with preference functions and weighted features (optimized by using GA).

As the controlling parameters of GA search for our experiments, we have used 1000 organisms in the population and set the crossover to 0.5 and mutation rate to 0.1. As a stopping condition, we have used 50 h, because it is noticed that after that time interval the solutions become stable.

For each CBR model, we applied the k-NN algorithm by varying parameter k ranging from 1 to 9. The 10 CV accuracies are illustrated in Tables 2–4. It can be noticed that the combination of CBR–PF–GA always shows better performance than the Pure CBR. This applies for the weighted model, as well.

We have further tested whether the detected differences are statistically significant. Wilcoxon signed ranks test is used to examine whether the classification performance of the hybrid approach is significantly higher than the classical one. Tables 5–7

Table 1
Datasets from the UCI repository.

Dataset	#Classes	#Instances	Nominal features	Numeric features	Total features
Australian credit	2	690	8	6	14
German credit	2	1000	13	7	20
SPSS credit	2	700	1	7	8

Table 2
Comparison of the models performance – Australian credit dataset.

Model	Classification accuracy – Australian credit dataset				
k	1	3	5	7	9
Pure CBR (%)	82.61	85.80	86.67	86.23	87.10
Pure CBR–PF–GA (%)	**82.90**	**87.54**	**87.97**	**88.12**	**87.83**
Weighted CBR (%)	82.61	86.81	86.09	87.25	86.96
Weighted CBR–PF–GA (%)	**85.22**	**88.41**	**88.55**	**88.26**	**88.26**

Table 3
Comparison of the models performance – German credit dataset.

Model	Classification accuracy – German Credit dataset				
k	1	3	5	7	9
Pure CBR (%)	71.60	72.80	73.0	73.30	73.10
Pure CBR–PF–GA (%)	**72.30**	**73.30**	**74.80**	**76.70**	**76.20**
Weighted CBR (%)	73.20	73.20	73.20	74.30	73.80
Weighted CBR–PF–GA (%)	**73.30**	**75.90**	**77.40**	**77.20**	**76.60**

Table 4
Comparison of the models performance – SPSS credit dataset.

Model	Classification accuracy – SPSS credit dataset				
k	1	3	5	7	9
Pure CBR (%)	72.71	74.86	74.86	76.00	77.29
Pure CBR–PF–GA (%)	**75.00**	**78.71**	**77.57**	**77.29**	**78.00**
Weighted CBR (%)	75.57	75.57	77.14	78.00	77.14
Weighted CBR–PF–GA (%)	**75.14**	**79.86**	**79.00**	**78.71**	**77.71**

Table 5
Wilcoxon signed ranks test – Australian credit dataset.

Pure CBR–PF–GA – Pure unweighted CBR					
k	1	3	5	7	9
Z	−2.11341	−0.77414	−1.80739	−2.25430	−0.56358
Asymp. sig. (2-tailed)	**0.03457**	0.43885	0.07070	**0.02418**	0.57304
Weighted CBR–PF–GA – Weighted CBR					
k	1	3	5	7	9
Z	−0.56358	−0.28109	−2.56516	−0.74200	−0.41208
Asymp. sig. (2-tailed)	0.57304	0.77864	**0.01031**	0.45809	0.68028

show the results of Wilcoxon signed ranks testing the classification ability of classical CBR model compared to CBR–PF–GA model. Unweighted models are compared separately from the weighted models.

For each dataset a *k* has been reported where CBR–PF–GA performed significantly better than the pure CBR at the 5% level. This indicates that the differences between results are in some cases significant. The similar situation is with the weighted models.

Table 6
Wilcoxon signed ranks test – German credit dataset.

Pure CBR–PF–GA – Pure unweighted CBR					
k	1	3	5	7	9
Z	−0.63402	−0.23820	−1.12437	−2.43935	−1.84192
Asymp. sig. (2-tailed)	0.52607	0.81173	0.26086	**0.01471**	0.06549
Weighted CBR–PF–GA – Weighted CBR					
k	1	3	5	7	9
Z	−0.16994	−1.60867	−2.19362	−2.25124	−2.44803
Asymp. sig. (2-tailed)	0.86505	0.10769	**0.02826**	**0.02437**	**0.01436**

Table 7
Wilcoxon signed ranks test – SPSS credit dataset.

Pure CBR–PF–GA – Pure unweighted CBR					
k	1	3	5	7	9
Z	−1.6999	−1.72083	−1.96338	−1.49671	−0.71630
Asymp. sig. (2-tailed)	0.28462	0.08528	**0.04960**	0.13447	0.47380
Weighted CBR–PF–GA – Weighted CBR					
k	1	3	5	7	9
Z	−0.23799	−2.03290	−1.40546	−1.01594	−0.35815
Asymp. sig. (2-tailed)	0.81189	**0.04206**	0.15989	0.30966	0.72023

4. Conclusions

We have suggested a new CBR model with a desire to improve the performance of the traditional CBR system. The new model represents a combination of CBR, preference theory functions and GA approach.

As preference functions provide more opportunities in expressing decision-maker preferences, they can be used in order to improve the case retrieval process.

We have used GA to optimize the parameters of preference functions, but also to optimize the importance (weights) of the features. For the experiments, we have chosen the fifth type of preference functions, as it is concluded that this type could be most useful since it covers the largest number of situations.

The experimental results clearly show that the proposed model may outperform the traditional CBR model.

However, this study has its limitations. Feature and instance selection are the factors that can be optimized in a CBR system. Furthermore, the computation of similarity for the categorical variables describing the cases can also be a topic for some future research. The choice of parameter k, representing the number of cases most similar to the new case, can also be used as the parameter for optimization. In this paper, we evaluated the proposed model for only for a number of different values of k, but it is also possible that some other values of *k* may improve the overall performance of the CBR system.

Finally, the generalization of the proposed model should be additionally tested in the future by applying it to the problems in other domains.

References

Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approach. *AI Communications, 7*(1), 39–59.

Ahn, H., & Kim, K.-J. (2008). Using genetic algorithms to optimize nearest neighbors for data mining. *Annals of Operations Research, 263*(1), 5–18.

Ahn, H., Kim, K.-J., & Han, I. (2007). A case-based reasoning system with the two-dimensional reduction technique for customer classification. *Expert Systems with Applications, 32*(4), 1011–1019.

Akhavein, J. D., Frame, W. S., & White, L. J. (2005). The diffusion of financial innovations: An examination of the adoption of small business credit scoring by large banking organisations. *The Journal of Business, 78*(2), 577–596.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–609.

Banasik, J., Crook, J., & Thomas, L. (2003). Sample selection bias in credit scoring models. *Journal of the Operational Research Society, 54*(8), 822–832.

Boritz, J. E., & Kennedy, D. B. (1995). Effectiveness of neural network types for prediction of business failure. *Expert Systems with Applications, 9*(4), 503–512.

Brans, J. P., & Vincke, Ph. (1985). A preference ranking organization method: The Promethee method for multiple criteria decision-making. *Management Science, 31*(6), 647–656.

Brans, J. P., Vincke, Ph., & Mareschal, B. (1986). How to select and how to rank projects: The Promethee method. *European Journal of Operational Research, 24*, 228–238.

Bryant, S. M. (1997). A case-based reasoning approach to bankruptcy prediction modeling. *International Journal of Intelligent Systems in Accounting, Finance and Management, 6*(3), 195–214.

Buta, P. (1994). Mining for financial knowledge with CBR. *AI Expert, 9*(2), 34–41.

Changchien, S. W., & Lin, M. C. (2005). Design and implementation of a case-based reasoning system for marketing plans. *Expert Systems with Applications, 28*(1), 43–53.

Chen, M. C., & Huang, S. H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications, 24*, 433–441.

Chen, Y.-K., Wang, C.-Y., & Feng, Y.-Y. (2010). Application of a 3NN+1 based CBR system to segmentation of the notebook computers market. *Expert Systems with Applications, 37*(1), 276–281.

Chiu, C. (2002). A case-based customer classification approach for direct marketing. *Expert Systems with Applications, 22*(2), 163–168.

Chiu, C., Chang, P. C., & Chiu, N. H. (2003). A case-based expert support system for due-date assignment in a water fabrication factory. *Journal of Intelligent Manufacturing, 14*(3–4), 287–296.

Chun, S. H., & Park, Y. J. (2006). A new hybrid data mining technique using a regression case based reasoning: Application to financial forecasting. *Expert Systems with Applications, 31*(2), 329–336.

Chye, K. H., Chin, T. W., & Peng, G. C. (2004). Credit scoring using data mining techniques. *Singapore Management Review, 26*(2), 25–47.

Coakley, J. R., & Brown, C. E. (2000). Artificial neural networks in accounting and finance. Modeling issues. *International Journal of Intelligent Systems in Accounting, Finance and Management, 9*(2), 119–144.

Desai, V. S., Convay, 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 Mathematics Applied in Business and Industry, 8*, 323–346.

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.

Dimitras, A. I., Zanakis, S. H., & Zopounidis, C. (1996). A survey of business failure with an emphasis on prediction methods and industrial applications. *European Journal of Operational Research, 90*(3), 487–513.

Elliott, R., & Filinkov, A. (2008). A self tuning model for risk estimation. *Expert Systems with Application, 34*(3), 1692–1697.

Frank, A., & Asuncion, A. (2010). *UCI machine learning repository* [<http://www.archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Fu, Y., & Shen, R. (2004). GA based CBR approach in Q&A system. *Expert Systems with Applications, 26*(2), 167–170.

Han, I., Chandler, J. S., & Liang, T. P. (1996). The impact of measurement scale and correlation structure on classification performance of inductive learning and statistical methods. *Expert Systems with Applications, 10*(2), 209–221.

Han, J., & Kamber, M. (2001). *Datamining: Concepts and techniques*. San Francisco, CA: Morgan Kaufmann Publishers.

Hsu, C. I., Chiu, C., & Hsu, P. L. (2004). Predicting information systems outsourcing success using a hierarchical design of case-based reasoning. *Expert Systems with Applications, 26*(3), 435–441.

Huang, C., Chen, M., & Wang, C. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications, 33*(4), 847–856.

Huang, J., Tzeng, G., & Ong, C. (2006). Two-stage genetic programming (2SGP) for the credit scoring model. *Applied Mathematics and Computation, 174*(2), 1039–1053.

Im, K. H., & Park, S. C. (2007). Case-based reasoning and neural network based expert system for personalization. *Expert Systems with Applications, 32*(1), 77–85.

Jo, H., & Han, I. (1996). Integration of case-based forecasting, neural network, and discriminant analysis for bankruptcy prediction. *Expert Systems with Applications, 11*(4), 415–422.

Kim, K. (2004). Toward global optimization of case-based reasoning systems for financial forecasting. *Applied Intelligence, 21*(3), 239–249.

Kim, K., & Han, I. (2001). Maintaining case-based reasoning systems using a genetic algorithms approach. *Expert Systems with Applications, 21*(3), 139–145.

Kolodner, J. L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan.

Kolodner, J. L., & Mark, W. (1992). Case-based reasoning. *IEEE Expert, 7*, 5–6.

Lee, T., & Chen, I. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*(4), 743–752.

Lee, T., Chiu, C., Lu, C., & Chen, I. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Applications, 23*(3), 245–254.

Leung, K., Cheong, F., & Cheong, C. (2007). Consumer credit scoring using an artificial immune system algorithm. In *Proceedings of the IEEE international conference on evolutionary computation (CEC 2007)* (pp. 3377–3384). IEEE Press. Singapore.

Li, H., & Sun, J. (2010). Business failure prediction using hybrid^2 case-based reasoning (H^{2}CBR). *Computers and Operations Research, 37*(1), 137–151.

Li, H., Sun, J., & Sun, B.-L. (2009). Financial distress prediction based on OR-CBR in the principle of k-nearest neighbors. *Expert Systems with Applications, 36*(1), 643–659.

Liao, T. W., Zhang, Z., & Mount, C. R. (1998). Similarity measures for retrieval in case-based reasoning systems. *Applied Artificial Intelligence, 12*, 267–288.

Lopez de Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., et al. (2005). Retrieval, reuse, revision, and retention in case-based reasoning. *Knowledge Engineering Review, 20*(3), 215–240.

Mareschal, B. (1986). Stochastic PROMETHEE multiple criteria decision making under uncertainty. *European Journal of Operational Research, 26*, 58–64.

Mareschal, B. (1988). Weight stability intervals in the PROMETHEE multicriteria decision aid method. *European Journal of Operational Research, 33*, 54–64.

Mareschal, B., & Brans, J. P. (1988). Geometrical representations for MCDM (GAIA). *European Journal of Operational Research, 34*, 69–77.

Martin, D. (1977). Early warning of bank failure: A logit regression approach. *Journal of Banking and Finance, 1*(3), 249–276.

Meyer, P. A., & Pifer, H. (1970). Prediction of bank failures. *The Journal of Finance, 25*, 853–868.

Montazemi, A. R., & Gupta, K. M. (1997). A framework for retrieval in case-based reasoning systems. *Annals of Operations Research, 72*, 51–73.

Park, C. S., & Han, I. (2002). A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction. *Expert Systems with Applications, 23*(3), 255–264.

Parreiras, R. O., & Vasconcelos, J. A. (2007). A multiplicative version of Promethee II applied to multiobjective optimization problems. *European Journal of Operational Research, 183*, 729–740.

Podvezko, V., & Podviezko, A. (2010). Dependence of multi-criteria evaluation result on choice of preference functions and their parameters. *Technological and Economic Development of Economy, 16*(1), 143–158.

Reinartz, T., Iglezakis, I., & Roth-Berghofer, T. (2001). Review and restore for case-based maintenance. *Computational Intelligence, 17*(2), 214–234.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Lawrence Erlbaum.

Shaw, M., & Gentry, J. (1998). Using an expert system with inductive learning to evaluate business loans. *Financial Management, 17*(3), 45–56.

Shin, K. S., & Han, I. (2001). A case-based approach using inductive indexing for corporate bond rating. *Decision Support Systems, 32*(1), 41–52.

Shin, K.-S., & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Applications, 16*(2), 85–95.

Slade, S. (1991). Case-based reasoning: A research paradigm. *AI Magazine, 12*(1), 42–55.

SPSS for Windows, Rel. 13.0. (2004). Chicago: SPSS Inc.

Thomas, L. C. (2000). A survey of credit and behavioral scoring – Forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*(2), 149–172.

Tseng, H. E., Chang, C. C., & Chang, S. H. (2005). Applying case-based reasoning for product configuration in mass customization environments. *Expert Systems with Applications, 29*(4), 913–925.

Wang, Y., & Ishii, N. (1997). A method of similarity metrics for structured representations. *Expert Systems with Applications, 12*(1), 89–100.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research, 27*(11–12), 1131–1152.

Wheeler, R., & Aitken, S. (2000). Multiple algorithms for fraud detection. *Knowledge-Based Systems, 13*(2), 93–99.

Yobas, M. B., Crook, J. N., & Ross, P. (2000). Credit scoring using evolutionary techniques. *IMA Journal of Mathematics Applied in Business & Industry, 11*(2), 111–125.

Zakrzewska, D. (2007). On integrating unsupervised and supervised classification for credit risk evaluation. *Information Technology and Control, 36*, 98–102.

Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and crossvalidation analysis. *European Journal of Operational Research, 116*(1), 16–32.