Applied Soft Computing 9 (2009) 599–607

Contents lists available at ScienceDirect

Applied Soft Computing

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

Bankruptcy prediction modeling with hybrid case-based reasoning and genetic algorithms approach

Hyunchul Ahn [a], Kyoung-jae Kim [b,*]

[a] School of Business IT, Kookmin University 861-1, Jeongneung-dong, Seongbuk-gu, Seoul 136-702, Korea
[b] Department of Management Information Systems, Dongguk University, 3-26 Pil-Dong, Chung-Gu, Seoul 100-715, South Korea

ARTICLE INFO

Article history:
Received 20 June 2007
Received in revised form 7 June 2008
Accepted 10 August 2008
Available online 26 August 2008

Keywords:
Case-based reasoning
Genetic algorithms
Feature weighting
Instance selection
Bankruptcy prediction

ABSTRACT

One of the most important research issues in finance is building effective corporate bankruptcy prediction models because they are essential for the risk management of financial institutions. Researchers have applied various data-driven approaches to enhance prediction performance including statistical and artificial intelligence techniques, and many of them have been proved to be useful. Case-based reasoning (CBR) is one of the most popular data-driven approaches because it is easy to apply, has no possibility of overfitting, and provides good explanation for the output. However, it has a critical limitation—its prediction performance is generally low. In this study, we propose a novel approach to enhance the prediction performance of CBR for the prediction of corporate bankruptcies. Our suggestion is the simultaneous optimization of feature weighting and the instance selection for CBR by using genetic algorithms (GAs). Our model can improve the prediction performance by referencing more relevant cases and eliminating noises. We apply our model to a real-world case. Experimental results show that the prediction accuracy of conventional CBR may be improved significantly by using our model. Our study suggests ways for financial institutions to build a bankruptcy prediction model which produces accurate results as well as good explanations for these results.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Prediction of corporate bankruptcies has long been an important topic and has been studied extensively in the finance and management literature because it is an essential basis for the risk management of financial institutions. Bankruptcy prediction models have used various statistical and artificial intelligence techniques. These techniques include discriminant analysis, logistic regression, decision tree, *k*-nearest neighbor, and artificial neural networks (ANNs) (see [1]). Among them, ANN has become one of the most popular techniques for the prediction of corporate bankruptcy due to its high prediction accuracy. ANN, however, has not been applied widely in financial companies because it is generally difficult to build models. The difficulty stems from many parameters to be set by heuristics. Furthermore, there is a danger of overfitting, and it is usually difficult to explain why it produces a specific result, i.e. poor explanation ability. So, there has been a need for other artificial intelligence techniques which have good explanation ability as well as high prediction performance.

Case-based reasoning (CBR) may be an alternative to relieve the above limitations of ANN. There is no possibility for overfitting because it uses specific knowledge of previously experienced problems rather than their generalized patterns [2]. Furthermore, CBR is maintained in an up-to-date state because the case-base is updated in real time, which is a very important feature for the real-world application.

Nevertheless, CBR has hardly attracted researchers' interest because its prediction accuracy is usually much lower than the accuracy of ANN. Thus, there have been many studies to enhance the performance of CBR. Among them, the mechanisms to enhance the case retrieval process such as the selection of the appropriate feature subsets, instance subsets and the determination of feature weights have been most frequently studied (see [3–7]).

One of the state-of-the-art techniques for CBR is simultaneous optimization of these parameters in CBR. Most prior research tried to optimize these parameters independently. However, we can find the global optimization model for CBR when considering these parameters simultaneously, which improves the prediction results synergetically.

* Corresponding author. Tel.: +82 2 910 4560; fax: +82 2 910 4519.
E-mail addresses: hyunchul.ahn@gmail.com (H. Ahn), kjkim@dongguk.edu, kjkim@dreamwiz.com (K.-j. Kim).

1568-4946/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2008.08.002

This study proposes a novel hybrid approach that optimizes the weights of the features and the training instances simultaneously by genetic algorithms (GAs). To validate the usefulness of our model, we apply it to the real-world case of corporate bankruptcy prediction and review the results produced by our model.

The rest of the paper is organized as follows. Section 2 briefly reviews prior studies, and Section 3 proposes our research model, the simultaneous optimization of feature weights and relevant instances by the GA approach. In the next section, the explanation for the research design and experiments are presented, and Section 5 describes all the empirical results and their meanings. In the final section, the conclusions of the study are presented.

2. Prior research

We review the prior studies on corporate bankruptcy prediction first. We also examine the general concept of CBR and the previous research to optimize it. After that, we review the recent studies regarding simultaneous optimization of several parameters for CBR systems. In the end, we examine the GA approach – the key method for simultaneous optimization – in detail.

2.1. Prior research on bankruptcy prediction using data-driven approaches

There has been substantial research into bankruptcy prediction because it is one of the most important problems for companies and financial institutions. Various techniques including ANN, decision tree, logistic regression, and discriminant analysis have been employed to predict corporate bankruptcy [1].

Early studies by Altman [8] and Deakin [9] used discriminant analysis to predict corporate bankruptcies. More recent research by Ohlson [10] used logit and probit models to predict bankruptcies. In addition, several studies in the past used artificial intelligence techniques to predict financial distress. In one of the earliest studies, Odom and Sharda [11] and Tam and Kiang [12] introduced ANN for predicting corporate bankruptcies. Following these studies, a number of studies further investigated the use of data mining techniques in financial distress prediction. Table 1 summarizes literature and methodological issues for using data mining techniques to predict corporate bankruptcies.

Table 1
Prior research on the prediction of corporate bankruptcies

Reference	Model	Benchmark models
Tam and Kiang [12]	BPN	DA, LR, *k*-NN, ID3
Martin-del-Brio and Serrano-Cinca [13]	SOM	N/A
Serrano-Cinca [14]	SOM	N/A
Serrano-Cinca [15]	BPN	DA, LR
Altman et al. [16]	BPN	DA
Wilson and Sharda [17]	BPN	DA
Boritz and Kennedy [18]	BPN	DA, LR, probit
Boritz et al. [19]	BPN	DA, *k*-NN, LR, Probit
Jo and Han [20]	BPN	DA, *k*-NN
Lee et al. [21]	BPN	LR, DA
Jo et al. [22]	BPN	DA, *k*-NN
Kiviluoto [23]	SOM, RBF-SOM, LVQ	DA, *k*-NN
Yang et al. [24]	PNN, BPN	DA
Zhang et al. [25]	BPN	LR
Shin and Lee [26]	GA	N/A
Shin et al. [27]	SVM	BPN

BPN, backpropagation neural networks; SOM, self-organizing map; RBF, radial basis function; LVQ, learning vector quantization; DA, discriminant analysis; LR, logistic regression; *k*-NN, *k*-nearest neighbor; PNN, probabilistic neural networks; GA, genetic algorithm; N/A, not applicable; SVM, support vector machines.

In Table 1, the authors of prior research mainly tested the feasibility of ANN in bankruptcy prediction. However, ANN has some limitations such as the possibility of overfitting the training data and its poor explanation ability for the results. Overcoming the danger of overfitting is crucial because bankruptcy prediction often needs huge data sets for generalizing the results. In addition, explanation ability is also important for real-world financial institutions in order to provide empirical evidence for decision makers. To address these limitations, this paper suggests a hybrid CBR and GA technique as a tool for financial distress prediction.

2.2. Case-based reasoning and optimization models

CBR is a problem solving technique that reuses past, similar cases to find solutions to problems. It provides a solution to a new problem or situation case by referencing a library of stored old cases—a case base. It mirrors the problem-solving approaches taken by human beings who solve current problems using past experiences. Most artificial intelligence approaches depends on general knowledge of a problem domain. However, CBR just refers to specific knowledge of previously experienced situations. Thus, it fits with complex and unstructured problems, and it is easy and convenient to update the knowledge base [4,28]. For these reasons, CBR has been popularly applied to management and engineering areas. Intelligent product catalogs for Internet shopping malls, conflict resolution in air traffic control, medical diagnosis and even the design of semiconductors are examples of CBR applications [29].

The process involved in CBR is represented by a 4-step cycle in Fig. 1 [30].

Among the steps of the above cycle, 'RETRIEVE' – the first step – is considered as the most important phase because the performance of CBR is determined here. The system matches a new problem against cases in the case base using a specific retrieval method, and finds the most similar cases in this step. This method is called 'nearest neighbor (NN) matching'. In NN matching, similar cases that are found affect the quality of the solution significantly, thus it is very important to design an effective retrieval method.

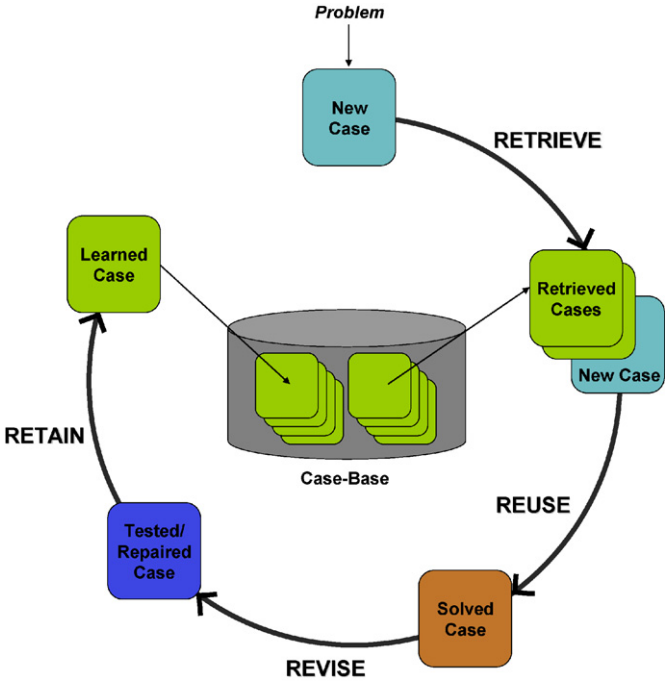

Fig. 1. Case-based reasoning cycle.

The similarity between an input case and stored cases can be determined in many ways. When cases are represented as feature vectors, calculating the weighted sum of feature distances is a common approach. Eq. (1) shows a typical numerical function for NN matching [31]:

$$\frac{\sum_{i=1}^{n} W_i \times \mathrm{sim}(f_i^{\mathrm{I}}, f_i^{\mathrm{R}})}{\sum_{i=1}^{n} W_i} \quad (1)$$

where W_i is the weight of the ith feature, f_i^{I} is the value of the ith feature for the input case, f_i^{R} is the value of the ith feature for the retrieved case, and sim( ) is the similarity function (usually, Euclidean distance) for f_i^{I} and f_i^{R}.

Eq. (1) contains many factors to be set in a heuristic way. There have been plenty studies to optimize them using scientific approaches. Among them, determining appropriate f_i (relevant features) and W_i (feature weights), and R (relevant instances) have been popular research topics in CBR literature.

2.3. Feature selection and weighting approaches in CBR

Feature selection is a method that uses only a small subset of features that prove to be relevant to the target concept. On the other hand, feature weighting is the method of assigning a proper weight to each feature according to its importance. Feature weighting can reflect the relative importance with sophistication, but feature selection can just determine whether the model would include a specific feature or not. That is, feature selection is a special case of feature weighting. Consequently, the prediction performance of the CBR system whose feature weights are optimized is always better than the CBR system whose feature selections are optimized.

There are many studies on feature selection. Stearns [32] proposed the *Sequential Forward Selection* (SFS) method which finds optimal feature subsets with the highest accuracy by varying the number of features. Siedlecki and Sklanski [33] proposed the genetic approach to feature subset selection and Cardie [34] used the decision tree method for a tool to select optimal features. Skalak [35] and Domingos [36] proposed different approaches for feature selection such as a hill climbing algorithm and a clustering method. In addition, Cardie and Howe [37] and Jarmulak et al. [31] suggested a combined model—the feature subset selection method and the feature weighting method. Their models selected relevant features using a decision tree in the first step, and then assigned weights to the selected features. The model from Cardie and Howe [37] determined the weights of the selected features using information gain, but Jarmulak et al. [31] used GA.

Kelly and Davis [38] proposed the GA approach to optimize feature weighting. Similar methods are applied to the prediction of corporate bond rating [4], failure-mechanism identification [39], and customer classification for customer relationship management [6]. Moreover, Wettschereck et al. [40] presented various feature weighting methods based on distance metrics in the machine learning literature and compared each method empirically.

2.4. Instance selection approaches

The instance selection technique has been proposed as a way of finding the representative cases in a case-base and determining a reduced subset of the case-base. Some of the literature calls this technique ‘editing’ or ‘prototype selection’. Reducing the whole case-base into a small subset that consists of only representative cases positively affects on conventional CBR systems. First of all, it reduces search space, so we can save computing time searching for nearest neighbors. It also produces quality results because it may eliminate noises in a case-base. Therefore, this issue has been researched for a long time, especially in computer science.

In the earliest study, Hart [41] proposed the condensed nearest neighbor algorithm and Wilson [42] presented *Wilson’s method*. Their primitive algorithms were based on simple information gain theory. Recently, researchers have applied mathematical tools or artificial intelligence techniques for instance selection. For example, Sanchez et al. [43] suggested the proximity graph approach and Lipowezky [44] presented a linear programming model as a tool for instance selection. In addition, Yan [45] and Huang et al. [46] proposed ANN to effectively select appropriate instances for CBR. Skalak [47] and Babu and Murty [48] suggested various schemes of GA approaches for instance selection and compared the performance of each method.

2.5. Simultaneous optimization approaches

Although prior research that proposed proper feature selection, feature weighting and instance selection might yield good results in CBR system, most previous studies tried to optimize these parameters independently. However, the simultaneous optimization model for CBR might improve the prediction results synergetically. Nevertheless, there are few studies on the simultaneous optimization of CBR due to its short history.

The first attempt to optimize feature selection and instance selection simultaneously was the study by Kuncheva and Jain [49]. They proposed the GA-based approach as an optimization tool and compared their model to sequential combining of conventional feature and instance selection algorithms. In their study, the results showed that their simultaneous optimization model outperformed other comparative models. After the pioneering work by Kuncheva and Jain [49], Rozsypal and Kubat [50] also proposed a similar model. However, they pointed out the model by Kuncheva and Jain [49] had defects when there are many training examples. Therefore, they used a different design for the chromosome and for the fitness function. They showed empirically that their model outperforms Kuncheva and Jain [49]. As an application research, Ahn et al. [51] applied the simultaneous optimization model to a customer classification problem, however there has been no study to apply it to bankruptcy prediction.

A point of clarification is that feature selection is a special case of feature weighting. It means the concept of feature weighting which varies the weights of features from 0 to 1 includes the concept of feature selection which is just binary selection, 0 or 1. Consequently, it is natural that the simultaneous optimization model for feature weighting and instance selection improves the performance of the model for feature selection and instance selection. In this manner, we can think of the simultaneous optimization model of feature weights and training instances as a mean to significantly enhance the performance of CBR.

Unfortunately, however, there have been few approaches to optimize feature weights and relevant instances simultaneously in case-based reasoning. Yu et al. [52] attempted simultaneous optimization of feature weighting and instance selection under a collaborative filtering (CF) environment. CF is the algorithm that is very similar to CBR because it also uses distance measure to determine the appropriate nearest neighbors. However, CF is not an algorithm for general purpose problem solving, but just for recommendation. Furthermore, Yu et al. [52] did not apply artificial intelligence techniques such as genetic algorithms, but an information-theoretic approach as a tool for optimization. Thus, their model was not the simultaneous optimization model but a sequential combining model of the two approaches in the strict sense of the word. Ahn et al. [53] tried to optimize feature weights

and instance selection simultaneously in a CBR to solve managerial problems. Their research model is very similar to the proposed model of this study. However, the application domains are completely different because their study applied the simultaneous optimization model to the problems of customer classification in online shopping malls.

2.6. Genetic algorithms for optimizing factors in case-based reasoning

As we reviewed in the previous section, GA is increasingly being used in CBR for finding optimum parameters. Table 2 summarizes some of the prior studies which try to optimize CBR using GA. As we can see from Table 2, there have been various approaches for optimizing the parameters for CBR except for the simultaneous optimization of feature weighting and instance selection. In general, there are few techniques like GA that enable the optimization of plural variables simultaneously from the global perspective. Thus, in this study, we also adopt GA as the search method of our simultaneous optimization model.

Genetic algorithms are adaptive search methods for finding optimal or near optimal solutions, premised on the evolutionary ideas of natural selection. The basic concept of GA is designed to simulate processes in the natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin in terms of the survival of the fittest. As such, they represent an intelligent exploitation of a random search within a defined search space to solve a problem. In general, the process of GA is as follows.

At first, GA generates the initial population randomly. In GA, population means a set of solutions, and each solution is called a chromosome. A chromosome has a form of binary strings in usual and all the parameters to be found are encoded on it. After generating the initial population, GA computes the fitness function of each chromosome. The fitness function is a user-defined function which returns the evaluation results of each chromosome, thus a higher fitness value means its chromosome is a dominant gene.

According to the fitness values, offspring are generated by applying genetic operators. In general, three operators are frequently used—reproduction, crossover, and mutation. By the reproduction operator, solutions with higher fitness values are reproduced with a higher probability. Crossover means exchanging substrings from pairs of chromosomes to form new pairs of chromosomes. The single point crossover, which separates chromosomes into two substrings, and the double point crossover, which separates them into three substrings, are the most popular crossover methods. Mutation involves generating mutations of the chromosomes. Mutation prevents the search process from falling into local maxima, but a mutation rate that is too high may cause great fluctuation. So, the mutation rate is generally set to a low value [54].

Applying these genetic operators and generating new generations of the population are repeated over and over until the stopping criteria are satisfied. In most cases, the stopping criterion is set to the maximum number of generations [6,55].

Table 2
Prior studies for CBR using GA

Reference	Optimized factors by GA		
	Feature selection	Feature weighting	Instance selection
Siedlecki and Sklanski [33]	O		
Jarmulak et al. [31]		O	
Kelly and Davis [38]		O	
Shin and Han [4]		O	
Liao et al. [39]		O	
Skalak [47]			O
Babu and Murty [48]			O
Kuncheva and Jain [49]	O		O
Rozsypal and Kubat [50]	O		O
Ahn et al. [51]	O		O
Ahn et al. [52]		O	O

3. Simultaneous optimization of feature weighting and instance selection using a genetic algorithm

This study proposes a novel CBR model whose feature weighting and instance selection are optimized globally, in order to improve prediction accuracy of typical CBR systems. Our model employs GA to select a relevant instance subset and to optimize the weights of each feature simultaneously using the reference and the test case-base. We call it *GOCBR* (*G*lobal *O*ptimization of feature weighting and instance selection using GA for *CBR*). The flowchart of *GOCBR* is shown in Fig. 2.

The detailed explanation for each step of *GOCBR* is presented as follows.

3.1. Phase 1. Initiation

In the first step, the system generates the initial population that would be used to find global optimum parameters—feature weights and selection variables for each instance. The values of the chromosomes for the population are initiated into random values before the search process. To enable GA to find the optimal parameters, we should design the structure of a chromosome, a form of binary strings. The structure of the chromosomes and population for *GOCBR* is represented in Fig. 3.

As shown in Fig. 3, each chromosome for *GOCBR* has all the information for feature weighting and instance selection. The length of each chromosome is $14 \times k + n$ bits when k is the number of features and n is the number of instances. In this study, we set the feature weights – ranging from 0 to 1 – as precise as 1/10,000. To preserve the precision level, 14 binary bits are required because $8192 = 2^{13} < 10{,}000 < 2^{14} = 16{,}384$ [56].

These encoded 14-bit binary numbers should be transformed into decimal floating numbers when the system needs to interpret the information contained in a chromosome. Eq. (2) shows the numeric transformation function for doing this job:

$$x' = \frac{x}{2^{14} - 1} = \frac{x}{16,383} \tag{2}$$

where x is the decimal number of the binary code for each feature weight.

For example, the binary code for feature 1 of the sample chromosome 1 in Fig. 3 is $(11111111111111)_2$. The decimal value of it is $(16{,}383)_{10}$ and it is interpreted as (16,383/16,383) = 1. The code for feature 1 of the sample chromosome 2 in Fig. 3 is $(10010011001001)_2$ whose decimal value of it is $(9417)_{10}$. It can be interpreted as $(9417/16{,}383) = 0.574806 \approx 0.5748$.

The value for the signs of instance selection is set to '0' or '1'. '0' means the corresponding instance is not selected and '1' means it is selected. n bits are required to implement instance selection by GA, where n is the number of total instances because the sign for instance selection needs just 1 bit.

3.2. Phase 2. Reasoning

After generating the initial population, the system performs a typical CBR process using the parameters in the chromosomes, and calculates the performance of each chromosome. The performance of each chromosome can be calculated through

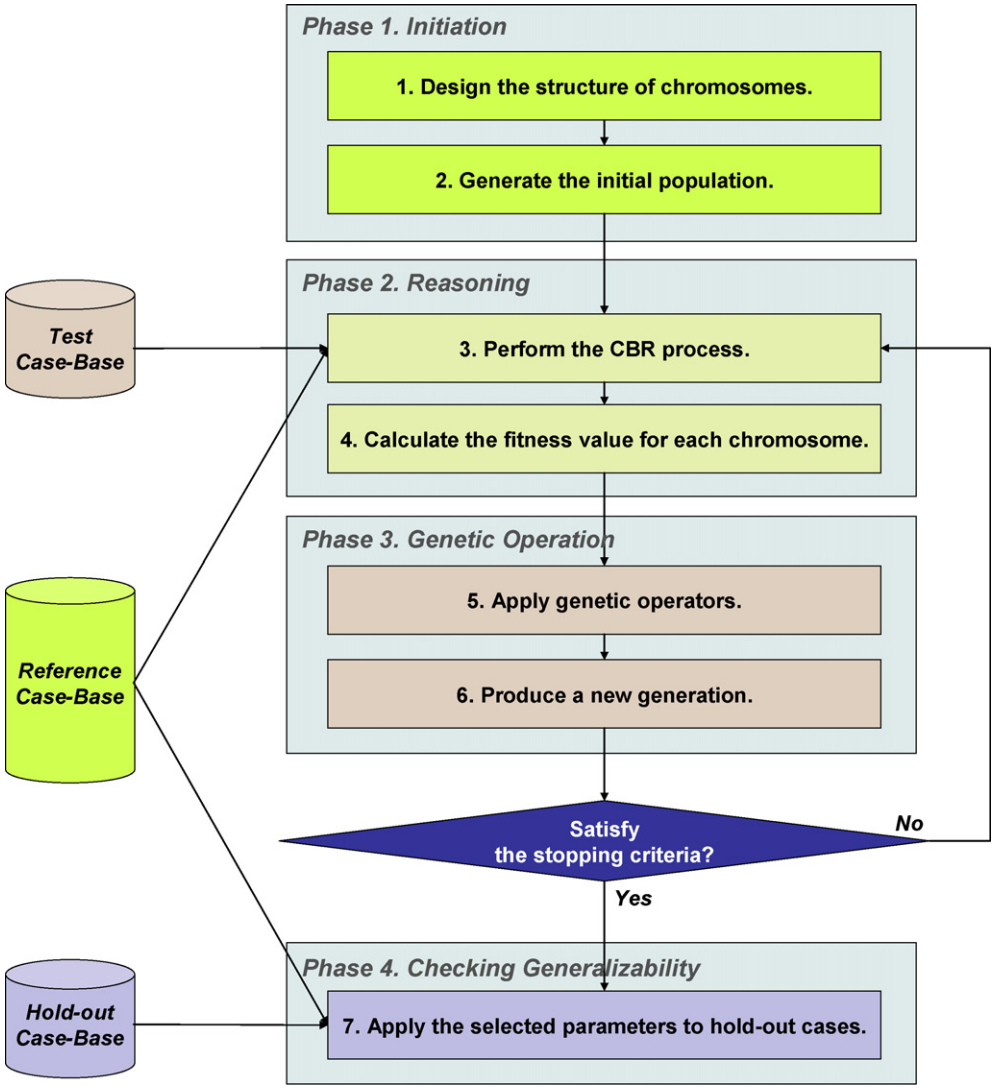

Fig. 2. Flowchart of *GOCBR*.

the fitness function for GA. In this study, the main goal is to find the optimal or near optimal parameters that produce the most accurate prediction solution. Thus, we set the fitness function (f_T) for the test data set T to the prediction accuracy of the test data set as in Eq. (3) [4,7,55]:

$$\text{Maximize} \quad f_T = \sum_{k=1}^{n} \text{hit}_k \tag{3}$$

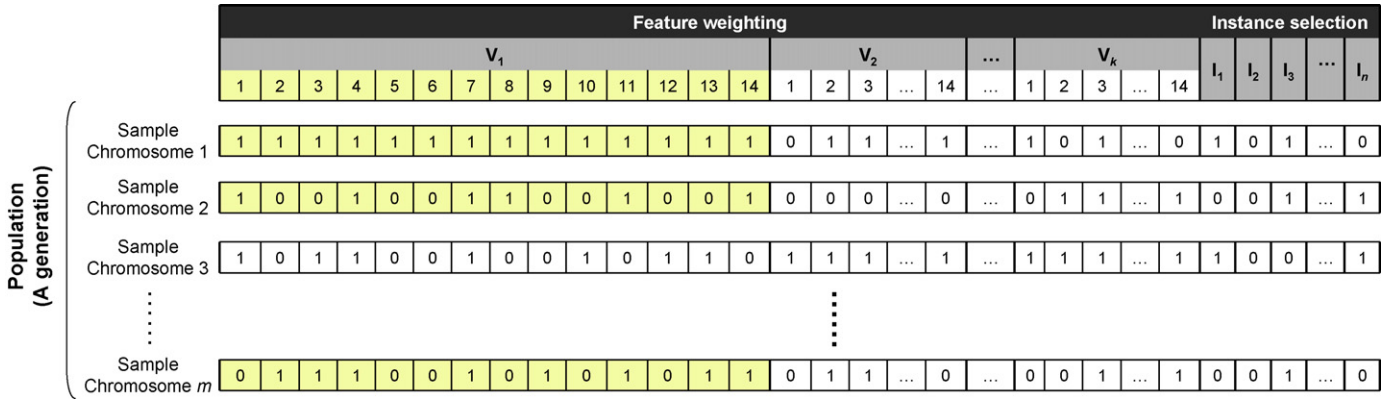

Fig. 3. Gene structure for *GOCBR*.

Table 3
Selected features and their statistics

Type	Name of feature	Range	Mean	Std. dev.	Wald	Sig.
Profitability	Financial expenses to liabilities	17.021	7.082	3.551	37.373	0.000
	Cost of sales to net sales	51.000	81.879	8.167	4.876	0.027
	Financial expenses and normal profit to total assets	156.723	17.054	23.104	178.444	0.000
	Financial expenses growth rate to assets	16.961	0.262	2.845	12.665	0.000
	Non-operating expenses growth rate to assets	24.991	−0.080	3.903	28.431	0.000
	Cost of sales × cost of sales growth ratio	1172.787	142.358	152.350	30.616	0.000
Liquidity	Solvency ratios	294.118	38.553	36.620	4.087	0.043
	Window coefficient	12.454	0.964	1.547	5.666	0.017
	Cash flow to total liabilities	3.021	0.094	0.323	16.127	0.000
Activity	Payables turnover	191.440	13.373	21.388	13.219	0.000
	Inventories growth rate to sales	51.987	1.597	7.146	9.660	0.002
	Total assets turnover × sales growth rate	12.352	1.991	1.837	18.751	0.000
Stability	Net worth to total assets	107.192	24.183	16.821	58.059	0.000
Growth	Total asset change ratios	111.592	20.390	20.739	54.823	0.000
Trend	Financial expenses growth	0.259	−0.007	0.035	4.289	0.038

where n is the size of the test data set T, hit_k is the matched result between the expected outcome (EO_k) and the actual outcome (AO_k), i.e. if $\text{EO}_k = \text{AO}_k$ then hit_k is 1, otherwise hit_k is 0.

3.3. Phase 3. Genetic operation

In the third step, a new generation of the population is produced by applying genetic operators such as reproduction, crossover, and mutation. According to the fitness values for each chromosome, the chromosomes whose values are high are selected and used for the basis of crossover. The mutation operator is also applied to the population with a very small mutation rate.

After the production of a new generation, phase 2 – the reasoning process with calculation of the fitness values – is performed again. From this point, phase 2 and phase 3 are iterated again and again until the stopping conditions are satisfied. When the stopping conditions are satisfied, the genetic search finishes and the chromosome which shows the best performance in the last population is finally selected as the final result.

3.4. Phase 4. Checking generalizability

Occasionally, the optimized parameters determined by GA fit with the test data very well, but they do not fit with the unknown data well. The phenomenon occurs when the parameters fit too well with the given test data set, i.e. overfitting. Thus, in the last stage, the system applies the finally selected parameters – the optimal weights of features and selection of instances – to the hold-out (unknown) data in order to check the generalizability of the parameters.

4. The research design and experiments

4.1. Application data

The application data used in this study consists of financial ratios and the status of bankrupt or non-bankrupt for corresponding corporations. The data is collected from one of the largest commercial banks in Korea. The sample consists of 1335 bankrupt companies in heavy industry which filed for bankruptcy between 1996 and 2000, and 1335 solvent companies in heavy industry between 1999 and 2000. Thus, the total number of samples is 2670 companies.

The financial status for each company is categorized as '0' or '1' and it is used as a dependent variable. '0' means that the company is bankrupt, and '1' means that the company is solvent. For independent variables, we first generate 164 financial ratios from the financial statement of each company. After that, we select 111 variables using two independent samples t-test. Finally, we choose 15 financial ratios as independent variables through the forward selection procedure based on logistic regression and the opinions of the experts who are responsible for approving and managing loans in a bank. Table 3 gives selected features and some statistics from outputs of descriptive statistics and logistic regression analysis.

4.2. Research design and system development

In order to validate the performance of the proposed model with sophistication, we experiment using five different CBR models for the same data set.

The first model is a typical CBR approach that does not have any mechanism to handle parameters. We label this model *TYCBR* (*TY*pical *CBR*). This model has no special process of feature subset selection or instance selection. Thus, all the features and instances are used for the reasoning process in this model. The relative importance of each feature is set equally, that is, it does not consider appropriate feature weights, either.

The second model, called *FSCBR* (*F*eature *S*election using GA for *CBR*), is the same as *TYCBR* except for the fact that it has a mechanism to optimize the selection of relevant features. In this model, it optimizes feature selection using GA. However, similar to *TYCBR*, it does not also consider optimal feature weights and relevant instances at all.

In the third model, GA finds not just optimal features, but the proper weight for each feature. As indicated before, weighting includes selection, so it provides the opportunity to enhance the performance of the model which uses just optimal selection. We name the model *FWCBR* (*F*eature *W*eighting using GA for *CBR*). *FWCBR* does not include instance selection, either.

The fourth model applies GA to choose an appropriate instance subset. We label it *ISCBR* (*I*nstance *S*election using GA for *CBR*). This model is unconcerned with feature selection or weighting. Thus, all features are selected and the weights for them are set equally.

The final model, called *FISCBR* (*F*eature and *I*nstance *S*election using the GA for *CBR*), is the two-dimensional simultaneous optimization model. It uses GA to find optimal relevant features

Table 4
The feature weights and instance selection of optimized CBR models

	FSCBR	*FWCBR*	*ISCBR*	*FISCBR*	*GOCBR*
Feature weights					
Financial expenses to liabilities	0	0.042650	1	1	0.429642
Cost of sales to net sales	1	0.191563	1	0	0.161996
Financial expenses and normal profit to total assets	1	0.874998	1	1	0.762192
Financial expenses growth rate to assets	0	0.150282	1	1	0.278488
Non-operating expenses growth rate to assets	1	0.962212	1	1	0.730993
Cost of sales × cost of sales growth ratio	1	0.450056	1	1	0.289991
Solvency ratios	1	0.471259	1	1	0.218830
Window coefficient	1	0.520006	1	1	0.339558
Cash flow to total liabilities	1	0.197122	1	1	0.100059
Payables turnover	1	0.800955	1	1	0.981829
Inventories growth rate to sales	0	0.345821	1	1	0.306552
Total assets turnover × sales growth rate	0	0.119975	1	1	0.083654
Net worth to total assets	1	0.136526	1	1	0.160738
Total asset change ratios	1	0.686527	1	1	0.847105
Financial expenses growth	0	0.135269	1	1	0.538748
Instance selections					
# of selections	1602	1602	1148	851	1445
Ratio (%)	100	100	71.66	53.12	90.20

and instances at the same time. This model is very similar to our proposed model, *GOCBR*. However, *GOCBR* optimizes feature weights rather than feature selection, which provides an opportunity to improve performance.

To apply these comparative models as well as our model, *GOCBR*, we developed a prototype system which provides the functions for *k*-NN (nearest neighbor) reasoning and GA optimization of the parameters for CBR. The base program for CBR was developed in Microsoft Excel 2003 using VBA (Visual Basic for Applications) and the function of GA optimization was implemented using Evolver Industrial version 4.06—a commercial GA tool. For the controlling parameters of the GA search in *GOCBR*, we use 100 chromosomes in the population and set the crossover rate to 70% and mutation rate to 10%. We set the stopping condition to 4000 trials (40 generations).

In addition, we also apply ANN to our data. We have mentioned that the motivation of the study is to build a bankruptcy prediction model that has not only explanation ability, but also performance as good as ANN. Thus, it is meaningful to check whether our proposed model has the prediction ability to serve as a substitute for ANN. To establish ANN, we adopt a standard three-layer back propagation network and set the learning rate to 0.1 and the momentum term to 0.1. The hidden and output nodes use sigmoid function as the transfer function. We perform the experiments repeatedly by varying the number of nodes in the hidden layer to 8, 16, 24 and 32. For the stopping criteria of ANNs, we allow 30,000 events since the minimum error.

5. Experimental results

5.1. The results of GA-optimized CBRs: FSCBR, FWCBR, ISCBR, FISCBR, and GOCBR

Table 4 shows the finally selected parameters of each model. As a result of *GOCBR*, we obtain 15 optimal weights of each feature and 1445 optimal training instances to maximize the prediction result for the test set. Because there are totally 1602 training samples, *GOCBR* selects about 90.26% from the total case base as an optimal instance subset. As we can see from Table 4, *GOCBR* selects more instances than *ISCBR* (71.66%) and *FISCBR* (53.12%).

The feature weights in Table 4 are not standardized, so direct comparison between the feature weights for each model is quite difficult. For this reason, we present Table 5 which shows the standardized weights of the features. As we can see from Table 5, the features for *FWCBR* have a slightly different pattern than the pattern for *GOCBR*. In the case of the variables related to 'cash flow' or 'liquidity', the weights for *FWCBR* are bigger than the ones of *GOCBR*. However, the opposite situation appears in the case of the variables that are related to 'financial expenses'. It may be interpreted that 'cash flow' plays an important role when

Table 5
The standardized feature weights of *FSCBR*, *FWCBR*, and *GOCBR*

Name of feature	*FSCBR*	*FWCBR*	*GOCBR*	Remarks
Financial expenses to liabilities	0.0000	0.0070	0.0690	*GOCBR* ≫ *FWCBR*
Cost of sales to net sales	0.1000	0.0315	0.0260	
Financial expenses & normal profit to total assets	0.1000	0.1438	0.1223	
Financial expenses growth rate to assets	0.0000	0.0247	0.0447	*GOCBR* ≫ *FWCBR*
Non-operating expenses growth rate to assets	0.1000	0.1581	0.1173	
Cost of sales × cost of sales growth ratio	0.1000	0.0695	0.0496	
Solvency ratios	0.1000	0.0774	0.0351	*FWCBR* ≫ *GOCBR*
Window coefficient	0.1000	0.0855	0.0545	
Cash flow to total liabilities	0.1000	0.0324	0.0161	*FWCBR* ≫ *GOCBR*
Payables turnover	0.1000	0.1316	0.1576	
Inventories growth rate to sales	0.0000	0.0534	0.0525	
Total assets turnover × sales growth rate	0.0000	0.0185	0.0143	
Net worth to total assets	0.1000	0.0224	0.0258	
Total asset change ratios	0.1000	0.1128	0.1360	
Financial expenses growth	0.0000	0.0222	0.0865	*GOCBR* ≫ *FWCBR*

Table 6
Average prediction accuracy of the models

Model	Test data set (%)	Hold-out data set (%)
TYCBR		80.75
FSCBR	82.58	82.06
FWCBR	83.90	83.93
ISCBR	83.71	82.62
FISCBR	84.64	83.17
GOCBR	87.08	86.73

Table 7
McNemar values for the hold-out data

Model	*FSCBR*	*FWCBR*	*ISCBR*	*FISCBR*	*GOCBR*
TYCBR	0.444	3.821*	3.115**	2.215	12.321***
FSCBR		1.227	0.048	0.284	6.063***
FWCBR			0.507	0.132	2.685**
ISCBR				0.056	5.513*
FISCBR					4.208*

* Significant at the 5% level.
** Significant at the 10% level.
*** Significant at the 1% level.

considering all data, but 'financial expenses' plays a more important role when considering only refined data in this data set.

5.2. Comparison of the prediction performances

Table 6 describes the prediction accuracy of each model which is produced when applying the parameters in Table 4. Among the models, *GOCBR* has the highest level of accuracy (86.73%) in the given hold-out data set, followed by *FWCBR* (83.93%), *FISCBR* (83.17%), *ISCBR* (82.62%), *FSCBR* (82.06%), and *TYCBR* (80.75%). The results show that *GOCBR* improves the prediction accuracy of typical CBR systems significantly by about 6% in this data set.

In order to examine whether the differences of predictive accuracy between *GOCBR* and other comparative algorithms are statistically significant, we apply the McNemar test to our experimental results. The McNemar test is a non-parametric technique to test the difference between paired proportions [4]. Table 7 shows the results of the McNemar tests to compare the performances of six algorithms for the hold-out data.

As shown in Table 7, *GOCBR* is better than *TYCBR* and *FSCBR* at the 1% level, and better than *ISCBR* and *FISCBR* at the 5% statistical significance level. *GOCBR* also outperforms *FWCBR* at the 10% statistical significance level.

In addition, we also use the two-sample test for proportions. This test may be used to determine whether two probabilities are the same. In this study, we apply it to determine if the hit ratios of the left-vertical methods are the same as the hit ratios of the right-horizontal methods [57]. Table 8 shows Z values for the pairwise comparison of performance between models. As shown in Table 8, *GOCBR* outperforms *TYCBR* at the 1% statistical significance level

Table 8
Z values for the hold-out data

Model	*FSCBR*	*FWCBR*	*ISCBR*	*FISCBR*	*GOCBR*
TYCBR	−0.550	−1.363*	−0.790	−1.034	−2.651**
FSCBR		−0.814	−0.240	−0.484	−2.106***
FWCBR			0.573	0.330	−1.296*
ISCBR				−0.244	−1.867***
FISCBR					−1.625*

* Significant at the 10% level.
** Significant at the 1% level.
*** Significant at the 5% level.

Table 9
The results of ANN

Number of hidden nodes	h = 8 (%)	h = 16 (%)	h = 24 (%)	h = 32 (%)
Training data set	88.01	87.88	86.01	86.51
Test data set	86.70	86.89	85.96	85.21
Hold-out data set	82.99	84.11	84.86	85.42

and also outperforms *FSCBR* and *ISCBR* at the 5% significance level. In addition, its performance is better than *FWCBR* and *FISCBR* at the 10% significance level. Table 8 also shows that *FWCBR* outperforms *TYCBR* at the 10% statistical significance level.

5.3. Comparison of GOCBR and ANN

In another comparative model, we apply ANN to our data set. Table 9 shows the performance of the ANN models whose number of the nodes in the hidden layer is 8, 16, 24, and 32 each.

As we can see above, the prediction accuracy of *GOCBR* (86.73%) is higher than the best performance of the ANN models (85.42%). However, the difference (1.31%) is not statistically significant when applying the McNemar test or two-sample test for proportions. However, the result can be interpreted as empirical proof that *GOCBR* may improve the prediction accuracy of conventional CBR up to the accuracy of ANN.

6. Conclusions

We have proposed a new hybrid CBR model using GA–*GOCBR*. Our proposed model optimizes feature weighting and instance selection simultaneously. By selecting optimal instances, it may reduce noises or distorted cases which lead erroneous prediction. Moreover, our model may also find appropriate nearest neighbors for CBR by applying optimal feature weights to similarity calculation, which may enhance the prediction accuracy. Compared to other models such as *TYCBR*, *FSCBR*, *FWCBR*, and *ISCBR* as well as *FISCBR*, *GOCBR* has the highest prediction accuracy in the empirical test for real-world bankruptcy prediction.

In bankruptcy prediction, ANN has been applied popularly because of its high prediction accuracy. Nevertheless, its limitations – overfitting and poor explanation ability of the results – have made people hesitate to use it as a method for bankruptcy prediction. Although CBR may overcome all of these limitations, its performance was weak compared to ANN in many prior studies (see [12,19,20,22,23]). However, the model proposed here has shown that well-optimized CBR may produce quality prediction results that are as good as ANN's. Thus, our study may provide new opportunities for using CBR as a tool for bankruptcy prediction of financial institutions.

However, there are some limitations in this study. First of all, the size of population and the number of generations for genetic search may be small when considering the size of search space. As a matter of fact, the search space for the simultaneous optimization of feature weighting and instance selection is very huge area, so it is necessary to extend the search space that is examined by GA. If we extend the search space of GA, our model – *GOCBR* – would be able to produce a more accurate prediction result.

Second, CBR models optimized by GA including *GOCBR* require too much time and computer resources. *GOCBR* iterates typical CBR process according to the evolving parameters during the GA process. A typical CBR process needs much computation because it should examine whole training case-base to make just one solution, so *GOCBR* is very time-consuming because it iterates typical CBR hundreds of thousands of times. Thus, future research should focus on ways to make *GOCBR* more efficient.

Third, we should consider other parameters to optimize CBR. For instance, the number of cases to combine – k parameter in k-NN – may be incorporated into our simultaneous optimization model [58,59]. The universal simultaneous optimization of feature weights and appropriate instances as well as other factors may upgrade the performance of CBR, although the search space for GA would require extension.

Finally, the generalizability of *GOCBR* should be tested in other problem domains. That is, whether *GOCBR* produces superior results in other applications should be validated. In this study, we apply the model to bankruptcy prediction. However, *GOCBR* can be applied to any other finance issue such as bond rating. Moreover, *GOCBR* can be applied to management issues such as the prediction of demand and supply, production quality, and even customers' behavior. Thus, *GOCBR* should be tested and validated further in other domains in the future.

References

[1] P.R. Kumar, V. Ravi, Bankruptcy prediction in banks and firms via statistical and intelligent techniques—a review, European Journal of Operational Research 180 (1) (2007) 1–28.
[2] I. Watson, Applying Case-based Reasoning: Techniques for Enterprise Systems, Morgan Kaufmann Publishers, San Francisco, CA, 1997.
[3] Y. Wang, N. Ishii, A method of similarity metrics for structured representations, Expert Systems with Applications 12 (1) (1997) 89–100.
[4] K.S. Shin, I. Han, Case-based reasoning supported by genetic algorithms for corporate bond rating, Expert Systems with Applications 16 (2) (1999) 85–95.
[5] K. Kim, I. Han, Maintaining case-based reasoning systems using a genetic algorithms approach, Expert Systems with Applications 21 (3) (2001) 139–145.
[6] C. Chiu, A case-based customer classification approach for direct marketing, Expert Systems with Applications 22 (2) (2002) 163–168.
[7] K. Kim, Toward global optimization of case-based reasoning systems for financial forecasting, Applied Intelligence 21 (3) (2004) 239–249.
[8] E.I. Altman, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, The Journal of Finance 23 (4) (1968) 589–609.
[9] E. Deakin, A discriminant analysis of predictors of business failure, Journal of Accounting Research 10 (1) (1974) 167–179.
[10] J. Ohlson, Financial ratios and the probabilistic prediction of bankruptcy, Journal of Accounting Research 18 (1) (1980) 109–131.
[11] M. Odom, R. Sharda, A neural network model for bankruptcy prediction, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN), San Diego, CA, (1990), pp. 163–168.
[12] K.Y. Tam, M.Y. Kiang, Managerial applications of the neural networks: the case of bank failure predictions, Management Science 38 (7) (1992) 926–947.
[13] B. Martin-del-Brio, C. Serrano-Cinca, Self-organizing neural networks for the analysis and representation of data: some financial cases, Neural Computing and Applications 1 (2) (1993) 193–206.
[14] C. Serrano-Cinca, Self organizing neural networks for financial diagnosis, Decision Support Systems 17 (3) (1996) 227–238.
[15] C. Serrano-Cinca, Feedforward neural networks in the classification of financial information, The European Journal of Finance 3 (3) (1997) 183–202.
[16] E.I. Altman, G. Macro, F. Varetto, Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks, Journal of Banking and Finance 18 (3) (1994) 505–529.
[17] R.L. Wilson, R. Sharda, Bankruptcy prediction using neural networks, Decision Support Systems 11 (5) (1994) 545–557.
[18] J.E. Boritz, D.B. Kennedy, Effectiveness of neural network types for prediction of business failure, Expert Systems with Applications 9 (4) (1995) 503–512.
[19] J.E. Boritz, D.B. Kennedy, A. Albuquerque, Predicting corporate failure using a neural network approach, Intelligent Systems in Accounting, Finance and Management 4 (2) (1995) 95–111.
[20] H. Jo, I. Han, Integration of case-based forecasting, neural network and discriminant analysis for bankruptcy prediction, Expert Systems with Applications 11 (4) (1996) 415–422.
[21] K.C. Lee, I. Han, Y. Kwon, Hybrid neural network models for bankruptcy predictions, Decision Support Systems 18 (1) (1996) 63–72.
[22] H. Jo, I. Han, H. Lee, Bankruptcy prediction using case-based reasoning, neural network and discriminant analysis, Expert Systems with Applications 13 (2) (1997) 97–108.
[23] K. Kiviluoto, Predicting bankruptcies with the self-organizing map, Neurocomputing 21 (1–3) (1998) 203–224.
[24] Z.R. Yang, M.B. Platt, H.D. Platt, Probabilistic neural networks in bankruptcy prediction, Journal of Business Research 44 (2) (1999) 67–74.
[25] G. Zhang, M.Y. Hu, B.E. Patuwo, D.C. Indro, Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis, European Journal of Operational Research 116 (1) (1999) 16–32.
[26] K.S. Shin, Y.J. Lee, A genetic algorithm application in bankruptcy prediction modeling, Expert Systems with Applications 23 (3) (2002) 321–328.
[27] K.S. Shin, T.S. Lee, H.J. Kim, An application of support vector machines in bankruptcy prediction model, Expert Systems with Applications 28 (1) (2005) 127–135.
[28] P. Humphreys, R. McIvor, F. Chan, Using case-based reasoning to evaluate supplier environmental management performance, Expert Systems with Applications 25 (2) (2003) 141–153.
[29] E. Turban, J.E. Aronson, Decision Support Systems and Intelligent Systems, 6th edition, Prentice-Hall, Upper Saddle River, NJ, 2001.
[30] A. Aamodt, E. Plaza, Case-based reasoning; foundational issues, methodological variations, and system approaches, AI Communications 7 (1) (1994) 39–59.
[31] J. Jarmulak, S. Craw, R. Rowe, Self-optimizing CBR Retrieval, in: Proceedings of the Twelfth IEEE International Conference on Tools with Artificial Intelligence, Vancouver, Canada, (2000), pp. 376–383.
[32] S. Stearns, On selecting features for pattern classifiers, in: Proceedings of the Third International Conference on Pattern Recognition, Coronado, CA, (1976), pp. 71–75.
[33] W. Siedlecki, J. Sklanski, A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters 10 (5) (1989) 335–347.
[34] C. Cardie, Using decision trees to improve case-based learning, in: Proceedings of the Tenth International Conference on Machine Learning, San Francisco, CA, (1993), pp. 25–32.
[35] D.B. Skalak, Prototype and feature selection by sampling and random mutation hill climbing algorithms, in: Proceedings of the Eleventh International Conference on Machine Learning, New Brunswick, NJ, (1994), pp. 293–301.
[36] P. Domingos, Context-sensitive feature selection for lazy learners, Artificial Intelligence Review 11 (1–5) (1997) 227–253.
[37] C. Cardie, N. Howe, Improving minority class prediction using case-specific feature weights, in: Proceedings of the Fourteenth International Conference on Machine Learning, San Francisco, CA, (1997), pp. 57–65.
[38] J.D.J. Kelly, L. Davis, Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm, in: Proceedings of the Fourth International Conference on Genetic Algorithms, San Diego, CA, (1991), pp. 377–383.
[39] T.W. Liao, Z.M. Zhang, C.R. Mount, A case-based reasoning system for identifying failure mechanisms, Engineering Applications of Artificial Intelligence 13 (2) (2000) 199–213.
[40] D. Wettschereck, D.W. Aha, T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, Artificial Intelligence Review 11 (1–5) (1997) 273–314.
[41] P.E. Hart, The condensed nearest neighbor rule, IEEE Transactions on Information Theory 14 (3) (1968) 515–516.
[42] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Systems, Man, and Cybernetics 2 (3) (1972) 408–421.
[43] J.S. Sanchez, F. Pla, F.J. Ferri, Prototype selection for the nearest neighbour rule through proximity graphs, Pattern Recognition Letters 18 (6) (1997) 507–513.
[44] U. Lipowezky, Selection of the optimal prototype subset for 1-NN classification, Pattern Recognition Letters 19 (10) (1998) 907–918.
[45] H. Yan, Prototype optimization for nearest neighbor classifier using a two-layer perceptron, Pattern Recognition 26 (2) (1993) 317–324.
[46] Y.S. Huang, C.C. Chiang, J.W. Shieh, E. Grimson, Prototype optimization for nearest-neighbor classification, Pattern Recognition 35 (6) (2002) 1237–1245.
[47] D.B. Skalak, Using a genetic algorithm to learn prototypes for case retrieval and classification, in: Proceedings of the 1993 AAAI Workshop on Case-based Reasoning, Washington, DC, (1993), pp. 64–69.
[48] T.R. Babu, M.N. Murty, Comparison of genetic algorithm based prototype selection schemes, Pattern Recognition 34 (2) (2001) 523–525.
[49] L.I. Kuncheva, L.C. Jain, Nearest neighbor classifier: simultaneous editing and feature selection, Pattern Recognition Letters 20 (11–13) (1999) 1149–1156.
[50] A. Rozsypal, M. Kubat, Selecting representative examples and attributes by a genetic algorithm, Intelligent Data Analysis 7 (4) (2003) 291–304.
[51] H. Ahn, K.-J. Kim, I. Han, A case-based reasoning system with the two-dimensional reduction technique for customer classification, Expert Systems with Applications 32 (4) (2007) 1011–1019.
[52] K. Yu, X. Xu, M. Ester, H.-P. Kriegel, Feature weighting and instance selection for collaborative filtering: an information-theoretic approach, Knowledge and Information Systems 5 (2) (2003) 201–224.
[53] H. Ahn, K.-J. Kim, I. Han, Global optimization of feature weights and the number of neighbors that combine in a CBR system, Expert Systems 23 (5) (2006) 290–301.
[54] J. Han, M. Kamber, Datamining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA, 2001.
[55] Y. Fu, R. Shen, GA based CBR approach in Q&A system, Expert Systems with Applications 26 (2) (2004) 167–170.
[56] Z.B. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd edition, Springer-Verlag, Berlin, Germany, 1996.
[57] D.L. Harnett, A.K. Soni, Statistical Methods for Business and Economics, Addison-Wesley, Massachusetts, MA, 1991.
[58] H. Ahn, K.-J. Kim, Using genetic algorithms to optimize nearest neighbors for data mining, Annals of Operations Research 163 (1) (2008) 5–18.
[59] H.Y. Lee, K.N. Park, Methods for Determining the optimal number of cases to combine in an effective case based forecasting system, Korean Journal of Management Research 27 (5) (1999) 1239–1252.