# Breast Cytology Diagnosis using a Hybrid Case-based Reasoning and Genetic Algorithms Approach

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

[a] *Center for Military Planning, Korea Institute for Defense Analyses*
*5-7, Cheongrangri 2-Dong, Dongdaemun-Gu, Seoul, 130-012, Korea*
*Tel: +82-2-961-1335, Fax: +82-2-961-1163, E-mail: hcahn@kida.re.kr*

[b] *Department of Management Information Systems, Dongguk University*
*3-26, Pil-Dong, Chung-Gu, Seoul, 100-715, Korea*
*Tel: +82-2-2260-3324, Fax: +82-2-2260-8824, E-mail: kjkim@dongguk.edu*

**Abstract**

*Case-based reasoning (CBR) is one of the most popular prediction techniques for medical diagnosis because it is easy to apply, has no possibility of overfitting, and provides a good explanation for the output. However, it has a critical limitation – its prediction performance is generally lower than other artificial intelligence techniques like artificial neural networks (ANNs). In order to obtain accurate results from CBR, effective retrieval and matching of useful prior cases for the problem is essential, but it is still a controversial issue to design a good matching and retrieval mechanism for CBR systems. In this study, we propose a novel approach to enhance the prediction performance of CBR. Our suggestion is the simultaneous optimization of feature weights, instance selection, and the number of neighbors that combine using genetic algorithms (GAs). Our model improves the prediction performance in three ways – (1) measuring similarity between cases more accurately by considering relative importance of each feature, (2) eliminating redundant or erroneous reference cases, and (3) combining several similar cases represent significant patterns. To validate the usefulness of our model, this study applied it to a real-world case for evaluating cytological features derived directly from a digital scan of breast fine needle aspirate (FNA) slides. Experimental results showed that the prediction accuracy of conventional CBR may be improved significantly by using our model. We also found that our proposed model outperformed all the other optimized models for CBR using GA.*

***Keywords:***

Case-based reasoning; Genetic algorithms; Feature weighting; Instance selection; The number of cases that combine; Breast cytology diagnosis

## Introduction

Case-based reasoning (CBR) is a reasoning technique that reuses past cases to find a solution to the new problem. The reasoning process of CBR is similar to the decision making process that human beings use in many real world applications. It often shows significant promise for improving the effectiveness of complex and unstructured decision making.

CBR has several strengths. In theory, there is no possibility of overfitting in CBR because it uses specific knowledge of previously experienced problems rather than their generalized patterns. In addition, 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. Also, it can explain why it provides a solution by presenting similar old cases. Consequently, it has been applied to various problem-solving areas including engineering, finance, marketing, and medical diagnosis. In particular, CBR is very appropriate for medical applications because the characteristics of CBR fit to medical domains very well. In usual, medical knowledge is incomplete, so medical applications put more stress on real cases than applications in other domains. In addition, the explanation capability of CBR can be more important in medical domains because it can be used as the helpful information source for decision makers (i.e. medical doctors).

Despite its various advantages, CBR has been criticized because its prediction accuracy is usually much lower than the accuracy of other artificial intelligence techniques, especially artificial neural networks (ANNs). 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 (Siedlecki & Sklanski, 1989; Cardie, 1993; Skalak, 1994; Domingos, 1997), instance subsets (Kelly & Davis, 1991; Wettschereck et al., 1997; Shin & Han, 1999; Liao et al., 2000; Chiu, 2002), the determination of feature weights (Skalak, 1993; Sanchez et al., 1997; Lipowezky, 1998; Yan, 1993; Babu & Murty, 2001; Huang et al., 2002), and the number of neighbors that combine (Lee and Park, 1999; Ahn et al., 2003) have been most frequently studied.

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 easily imagine that the global optimization model for CBR which considers these parameters simultaneously may improve the prediction results.

This study proposes a novel hybrid approach that optimizes three parameters of CBR simultaneously by genetic algorithms (GAs) – (1) the weights of the features, (2) the training instances, and (3) the number of neighbor cases that combine. To validate the usefulness of our model, we apply it to the real-world case of breast cytology diagnosis via digital image analysis.

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, relevant instances and the number of neighbors that combine 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.

## Prior Research

### 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. These characteristics of CBR make it appropriate for diagnosis, prognosis and prescription in medical areas.

The process involved in CBR is represented by a 4-step cycle in Figure 1 (Aamodt & Plaza, 1994). Among the steps of CBR, '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. 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.

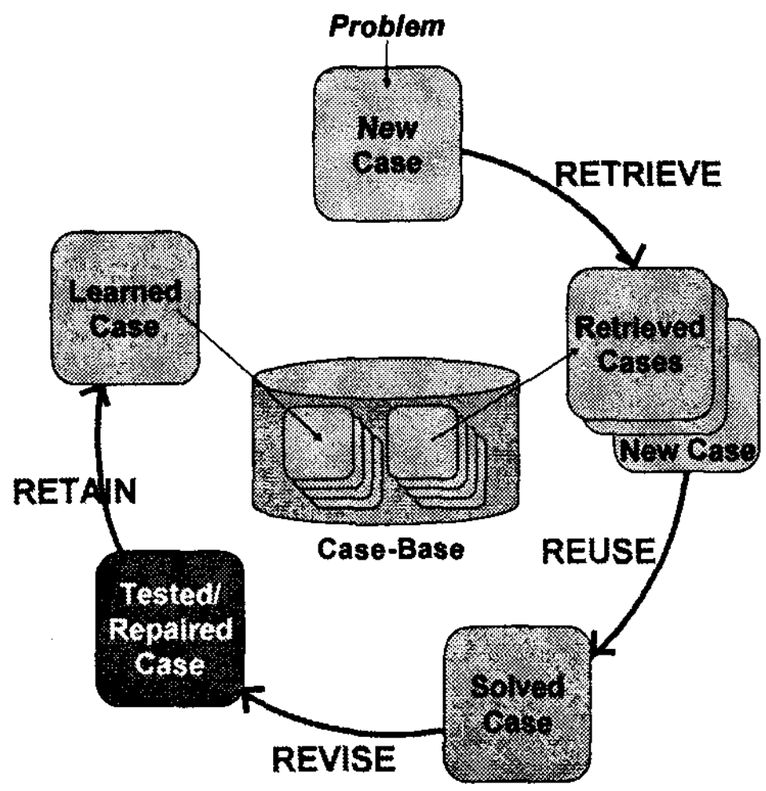


*Figure 1 – Case-based reasoning cycle*

Equation (1) shows a typical numerical function for NN matching (Jarmulak et al., 2000).

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

where $W_i$ is the weight of the $i$ th feature, $f_i^I$ is the value of the $i$ th feature for the input case, $f_i^R$ is the value of the $i$ th feature for the retrieved case, and *sim()* is the similarity function (usually, Euclidean distance) for $f_i^I$ and $f_i^R$.

Equation (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.

### 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 (1976) 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 (1989) proposed the genetic approach to feature subset selection and Cardie (1993) used the decision tree method for a tool to select optimal features. Skalak (1994) and Domingos (1997) proposed different approaches for feature selection such as a hill climbing algorithm and a clustering method. In addition, Cardie and Howe (1997) and Jarmulak et al. (2000) 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 (1997) determined the weights of the selected features using information gain, but Jarmulak et al. (2000) used GA.

Kelly and Davis (1991) proposed the GA approach to optimize feature weighting. Similar methods are applied to the prediction of corporate bond rating (Shin and Han, 1999), failure-mechanism identification (Liao et al., 2000), and customer classification for customer relationship management (Chiu, 2002). Moreover, Wettschereck et al. (1997) presented various feature weighting methods based on distance metrics in the machine learning literature and compared each method empirically.

**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. 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 (1968) proposed the condensed nearest neighbor algorithm and Wilson (1972) 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. (1997) suggested the proximity graph approach and Lipowezky (1998) presented a linear programming model as a tool for instance selection. In addition, Yan (1993) and Huang et al. (2002) proposed ANN to effectively select appropriate instances for CBR. Skalak (1993) and Babu and Murty (2001) suggested various schemes of GA approaches for instance selection and compared the performance of each method.

**Optimization of the Number of Neighbors that Combine**

Regarding case retrieval, many CBR systems use the one-nearest neighbor (1-NN) method. It's the method for retrieving the most similar case from the case-base and make predictions based on it. However, to improve performance, some CBR systems retrieve several similar cases simultaneously and make predictions by combining these all cases (e.g. voting or interpolation). This is called $k$ nearest neighbor ($k$-NN) retrieval. The parameter, $k$, means the number of cases that combine. Values of $k$ larger than 1 may be used to improve the generalization properties of the retrieval, and reduce the sensitivity to noise. That is, a large $k$ parameter may improve the accuracy of CBR prediction results. However, if $k$ is too large, the prediction accuracy may be lower because the selected similar cases would include many noisy cases. Thus, finding the optimal $k$ parameter for $k$-NN is also important in order to improve the accuracy of these kinds of retrieval systems. Nonetheless, there are few studies that tried to optimize it.

Lee and Park (1999) proposed three methods for optimizing the number of cases that combine. They are (1) fixing the number of cases that combine (conventional $k$-NN), (2) optimal spanning methods, and (3) the mathematical programming (MP) model using similarity distribution. A simulation study was conducted to test the performance of each model, and it proved that the MP model using similarity distribution was the best among those suggested. Equation (2) shows the objective function and constraints in their MP model:

$$Max.\ \ \mathrm{SF} = \frac{\sum_{b=1}^{n} S_{tb} Z_b}{\left(\sum_{b=1}^{n}\sum_{q=1}^{n} S_{bq} Z_b Z_q\right)^p} \qquad (2)$$

$$s.t.\quad (S_{tb} - S_{tq}) \times (Z_b - Z_q) \geq 0 \quad \forall\, b, q$$

$$Z_b = 0 \text{ or } 1$$

$$0 \leq p \leq 0.5$$

where $n$ is the total number of cases in a reference case-base, $S_{tb}$ is the similarity between target case (input case) $t$ and base case (retrieved case) $b$, and $s_{bq}$ is the similarity between base case $b$ and another base case $q$. Finally, $Z_b$ is the binary sign variable which represents whether base case $b$ is selected or not.

Their MP model is worthwhile because it is the first attempt to optimize the $k$ parameter, and it is based on concrete science including linear programming (LP) and statistics. However, their suggestion has several critical limitations. First of all, we can infer from the above equations that the optimal number of cases that combine wholly depends on each input case $t$. That is, the model computes a different optimal $k$ every time it gets a new input case. Thus, this model may not suggest the optimal value of the $k$ parameter that can be applied generally, and it also causes too much computation time that may disable real-time prediction. Furthermore, this model still has a variable to optimize, parameter $p$. The authors explain parameter $p$ as an adjusting factor to determine the number of cases that combine, but there is not a precise definition for parameter $p$. In addition, they do not suggest a mechanism for determining the appropriate value of parameter $p$.

To mitigate the limitations of Lee and Park (1999), Ahn et

al. (2003) suggests a new optimization model for *k* parameter, which applies a genetic algorithm (GA) as a tool for optimizing *k* parameter of *k*-NN. Park et al. (2006) also proposes a novel CBR model called PCBR (Probabilistic CBR), which optimizes the nearest neighbors by using statistical distribution of distances between cases.

**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 (1999). 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 (1999), Rozsypal and Kubat (2003) also proposed a similar model. However, they pointed out the model by Kuncheva and Jain (1999) 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 (1999).

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. Yu et al. (2003) attempted simultaneous optimization of feature weighting and instance selection using an information-theoretic approach under a collaborative filtering (CF) environment, which is very similar to CBR. Ahn et al. (2006) also proposed the simultaneous optimization model of feature weighting and instance selection using GA for CBR. However, unfortunately, there has been no approach to optimize feature weights, relevant instances, and the number of neighbors that combine simultaneously in case-based reasoning, as far as we know.

**Genetic Algorithms for Optimizing Factors in CBR**

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.

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 (Han & Kamber, 2001; Chiu, 2002; Fu & Shen, 2004).

As we reviewed in the previous sections, GA is increasingly being used in CBR for finding optimal parameters. 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.

## Global Optimization of CBR using GA

This study proposes a novel CBR model whose feature weighting, instance selection, and *k* parameter of *k*-NN 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 and the number of neighbors that combine simultaneously using the reference and the test case-base. We call it *GOCBR* (*G*lobal *O*ptimization of feature weights, instance selection, and the number of neighbors that combine using GA for *CBR*). The flowchart of *GOCBR* is shown in Figure 2.

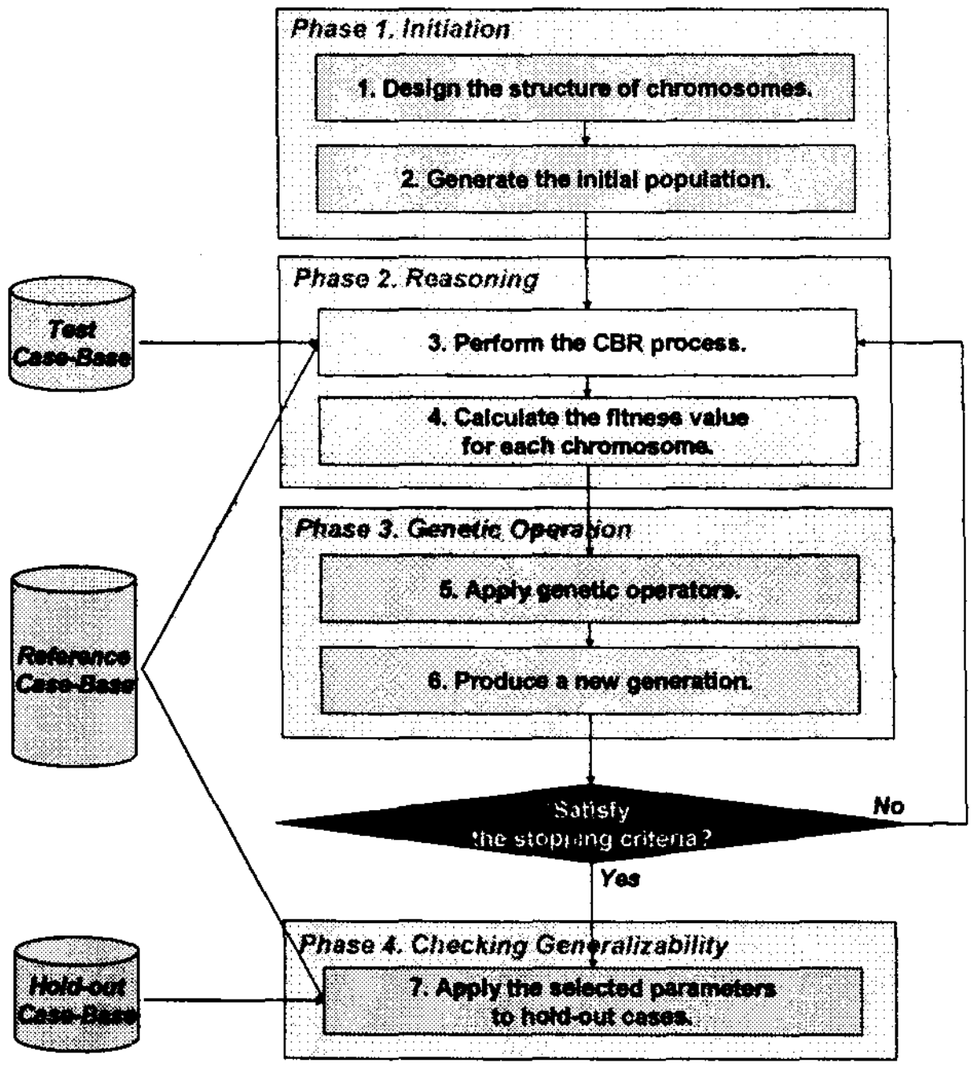


*Figure 2 – Flowchart of GOCBR*

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

***Phase 1. Initiation***

In the first step, the system generates the initial population that would be used to find global optimum or near-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 or near-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 Figure 3.

As shown in Figure 3, each chromosome for *GOCBR* has all the information for feature weighting, instance selection, and *k* parameter. The length of each chromosome is $3\times m+n+3$ bits when $m$ is the number of features and $n$ is the number of instances.

Here, we encode the feature weights as the relative importance of each feature in a 7-point scale (0: useless, 7: very important). That is, they range from 0 to 7 (totally, 8 states), so 3 binary bits are required for each feature because $2^3=8$. These 3-bit binary numbers are transformed into decimal floating weights, which range from 0 to 1 by applying the following Equation (3).

$$w_i = \frac{x_i}{\sum_{i=1}^{N} x_i} \tag{3}$$

where $w_i$ is the feature weight for feature $i$, $N$ is the number of total features, and $x_i$ is the decimal number of the binary code for the relative importance of feature $i$.

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.

The value of the $k$ parameter may be differently encoded according to the size of the search space. In this study, GA searches for $k$ from 1 to 8 ($=2^3$), so 3 bits are devoted for the $k$ parameter.

***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 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 Equation (4) (Shin & Han, 1999; Fu & Shen, 2004; Kim, 2004).

$$\text{Maximize } f_T = \sum_{k=1}^{n} hit_k \tag{4}$$

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 $EO_k=AO_k$ then $hit_k$ is 1, otherwise $hit_k$ is 0.

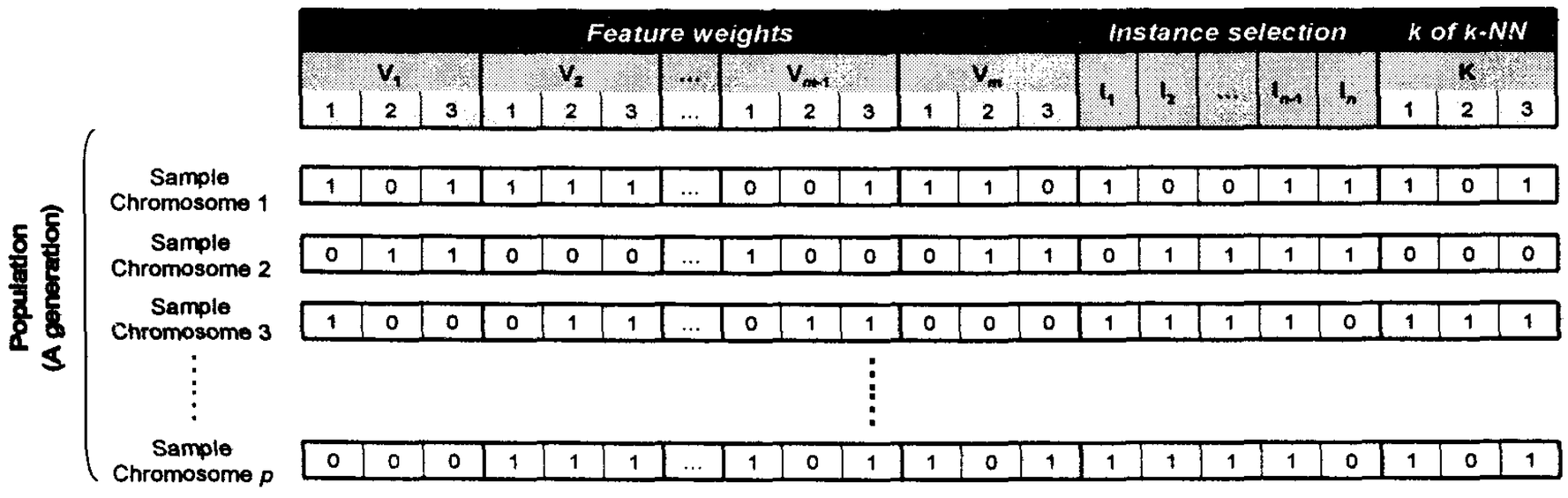


*Figure 3 – Gene structure for GOCBR*

***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.

***Phase 4. Checking Generalizability***

Occasionally, the optimized parameters determined by GA fit with the test data very well, but they don't 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, selection of instances, and *k* parameter of *k*-NN – to the hold-out (unknown) data in order to check the generalizability of the parameters.

## The Research Design and Experiments

### Application Data

In general, there are three available methods for diagnosing breast cancer: mammography, fine needle aspirate (FNA) with visual interpretation, and surgical biopsy. Among them, surgical biopsy is known to be the most accurate method, however it is invasive, time consuming, and costly. Thus, diagnosis systems based on digital image analysis that allow an accurate diagnosis without the need for a surgical biopsy are considered as a realistic alternative. FNA involves using a small gauge needle to take the fluid directly from a breast lump or mass, previously detected by self-examination and/or mammography. The fluid from the FNA is placed on a glass slide and stained to highlight the nuclei of the constitute cells. An image from the FNA is transferred to a workstation by a video camera mounted on a microscope (Estévez et al., 2002).

The database used for this research was taken from the Wisconsin Breast Cancer Database, which is one of the public medical databases provided by UCI repository (Blake & Merz, 1998). It is an image database for classification algorithm testing, made publicly available by Dr. William H. Wolberg of the University of Wisconsin Hospitals. The database consists of a personal series (WHW) of 569 consecutive breast aspirates that contained epithelial cells (212 with cancer, and 357 with fibrocystic disease). The database totally provides 30 independent features that are extracted from the image. These features consist of the (1) mean, (2) standard error, and (3) "worst" or largest (mean of the three largest values) of the 10 variables: radius (mean of distances from center to points on the perimeter), texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter$^2$ ÷ area - 1.0), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, and fractal dimension. Using this image database, we try to apply our proposed model to help to this diagnosis problem.

The final diagnosis result for each image is categorized as '0' or '1' and it is used as a dependent variable. '0' means that the cell is benign, and '1' means that the cell is malignant. For independent variables, we select 25 variables from 30 variables using two independent samples t-test with a confidence level of 95%. And, we apply zero-mean normalization. By using this type of normalization, the mean of the transformed set of data points is reduced to zero. For this, the mean and standard deviation of the initial set of data values are required. Zero-mean normalization maps value $v$ of $A$ to $v'$ by computing Equation. (5).

$$v' = \frac{v - \mu_A}{\sigma_A} \tag{5}$$

where $\mu_A$ and $\sigma_A$ are the mean and standard deviation of the initial data values (i.e. set $A$).

Zero-mean normalization is usually employed to enhance the performance of the CBR systems because it ensures that larger value input features do not overwhelm smaller value input features (Han & Kamber, 2001).

In our experiment, the data is split into the three groups: reference, test and hold-out case-bases. The reference case-base is used to search for the optimal feature weights, instance selection, and k parameter in genetic learning, and it is also used as a case-base for retrieval. The test case-base is used for generalization by balancing the results from training and test data. The final one, the hold-out case-base, is used for evaluating prediction accuracy. The number of cases in each case-base is shown in Table 1.

*Table 1 - The portion of each case-base*

| *Case-base* | *Portion* | *Benign cases* | *Malignant cases* | *Total* |
|---|---|---|---|---|
| Reference | 60% | 215 | 128 | 343 |
| Test | 20% | 71 | 42 | 113 |
| Hold-out | 20% | 71 | 42 | 113 |
| Total | 100% | 357 | 212 | 569 |

### Research Design and System Development

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

The first model is a typical CBR approach that doesn't 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 doesn't consider appropriate feature weights, either. In this model, the $k$ parameter of $k$-NN is set at 1.

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 doesn't also consider optimal feature weights, relevant instances, and the number of neighbors that combine 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* doesn't include instance selection or optimization of $k$ parameter of $k$-NN, 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. Here, $k$ parameter of $k$-NN is also set at 1.

The fifth 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 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. In addition, this model doesn't consider the number of neighbors that combine.

The final model, called *FWISCBR* (*F*eature *W*eighting and *I*nstance *S*election using the GA for *CBR*), is the extended simultaneous optimization model of *FISCBR*. It uses GA to find optimal relevant features weights and instances at the same time. This model is almost same to our proposed model except that it sets $k$ parameter of $k$-NN at 1 rather than optimizing it.

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 and the function of GA optimization was implemented using Evolver Industrial version 4.08. For the controlling parameters of the GA search in *GOCBR*, we use 200 chromosomes in the population and set the crossover rate to 70% and mutation rate to 10%. We set the stopping condition to 4000 trials (20 generations).

## Experimental Results

### The Results of GA-optimized CBRs

Table 2 shows the finally selected parameters of each model. As a result of GOCBR, we obtain 25 optimal weights of each feature and 176 optimal training instances to maximize the prediction result for the test set. Because there are totally 343 training samples, GOCBR selects about 51.31% from the total case base as an optimal instance subset. As we can see from Table 2, GOCBR selects fewer instances than FISCBR (51.60%) and FWISCBR (79.02%), but it selects more instances than ISCBR (47.52%).

*Table 2 - The portion of each case-base*

| | *TY CBR* | *FS CBR* | *FW CBR* | *IS CBR* | *FIS CBR* | *FWIS CBR* | *GO CBR* |
|---|---|---|---|---|---|---|---|
| ***Feature weights*** | | | | | | | |
| M_RAD | 0.040 | 0.143 | 0.071 | 0.040 | 0.000 | 0.073 | 0.047 |
| M_TXTR | 0.040 | 0.143 | 0.000 | 0.040 | 0.111 | 0.000 | 0.093 |
| M_PERI | 0.040 | 0.000 | 0.024 | 0.040 | 0.000 | 0.024 | 0.023 |
| M_AREA | 0.040 | 0.000 | 0.024 | 0.040 | 0.000 | 0.024 | 0.070 |
| M_SMOT | 0.040 | 0.000 | 0.071 | 0.040 | 0.000 | 0.073 | 0.070 |
| M_COMP | 0.040 | 0.000 | 0.000 | 0.040 | 0.000 | 0.049 | 0.023 |
| M_COCA | 0.040 | 0.000 | 0.000 | 0.040 | 0.000 | 0.000 | 0.070 |
| M_CPNT | 0.040 | 0.000 | 0.071 | 0.040 | 0.111 | 0.073 | 0.070 |
| M_SYMM | 0.040 | 0.000 | 0.000 | 0.040 | 0.000 | 0.000 | 0.000 |
| SE_RAD | 0.040 | 0.000 | 0.071 | 0.040 | 0.000 | 0.049 | 0.047 |
| SE_PERI | 0.040 | 0.000 | 0.095 | 0.040 | 0.000 | 0.024 | 0.070 |
| SE_AREA | 0.040 | 0.143 | 0.000 | 0.040 | 0.111 | 0.000 | 0.023 |
| SE_COMP | 0.040 | 0.000 | 0.048 | 0.040 | 0.000 | 0.024 | 0.000 |
| SE_COCA | 0.040 | 0.000 | 0.000 | 0.040 | 0.000 | 0.000 | 0.000 |
| SE_CPNT | 0.040 | 0.000 | 0.048 | 0.040 | 0.000 | 0.073 | 0.000 |
| WR_RAD | 0.040 | 0.143 | 0.071 | 0.040 | 0.111 | 0.098 | 0.047 |
| WR_TXTR | 0.040 | 0.000 | 0.024 | 0.040 | 0.111 | 0.073 | 0.070 |
| WR_PERI | 0.040 | 0.000 | 0.048 | 0.040 | 0.000 | 0.073 | 0.000 |
| WR_AREA | 0.040 | 0.143 | 0.095 | 0.040 | 0.111 | 0.049 | 0.023 |
| WR_SMOT | 0.040 | 0.143 | 0.095 | 0.040 | 0.111 | 0.073 | 0.093 |
| WR_COMP | 0.040 | 0.000 | 0.024 | 0.040 | 0.000 | 0.024 | 0.023 |
| WR_COCA | 0.040 | 0.000 | 0.024 | 0.040 | 0.111 | 0.073 | 0.023 |
| WR_CPNT | 0.040 | 0.000 | 0.024 | 0.040 | 0.000 | 0.024 | 0.023 |
| WR_SYMM | 0.040 | 0.000 | 0.000 | 0.040 | 0.000 | 0.000 | 0.023 |
| WR_FRAC | 0.040 | 0.143 | 0.071 | 0.040 | 0.111 | 0.024 | 0.070 |
| ***Selected features*** | 25 | 7 | 18 | 25 | 9 | 19 | 20 |
| ***Selected instances*** | 343 | 343 | 343 | 163 | 177 | 271 | 176 |
| ***k parameter of k-NN*** | 1 | 1 | 1 | 1 | 1 | 1 | 3 |

### Comparison of the Prediction Performances

Table 3 describes the prediction accuracy of each model which is produced when applying the parameters in Table 2. Among the models, GOCBR has the highest level of accuracy (99.12%) in the given hold-out data set, followed by FWISCBR (98.23%), FISCBR, ISCBR, FWCBR (97.35%), FSCBR (96.46%), and TYCBR (95.58%). The results show that GOCBR improves the prediction accuracy of typical CBR systems slightly by about 3.54% in this data set.

*Table 3 - Average prediction accuracy of the models*

| Model | Test data set | Hold-out data set |
|---|---|---|
| *TYCBR* | | 95.58% |
| *FSCBR* | 97.35% | 96.46% |
| *FWCBR* | 98.23% | 97.35% |
| *ISCBR* | 98.23% | 97.35% |
| *FISCBR* | 98.23% | 97.35% |
| *FWISCBR* | 98.23% | 98.23% |
| *GOCBR* | 99.12% | 99.12% |

## Conclusions

We have proposed a new hybrid CBR model using GA – GOCBR. Our proposed model optimizes feature weighting, instance selection, and the number of neighbors that combine simultaneously. By selecting optimal instances, it may reduce noises or distorted cases which lead erroneous prediction. Our model may also find appropriate nearest neighbors for CBR by applying optimal feature weights to similarity calculation, which may enhance the prediction accuracy. In addition, it generates prediction results by referencing appropriate number of similar cases, which represent the inherent patterns with minimization of external errors. Compared to other models such as TYCBR, FSCBR, FWCBR, ISCBR, FISCBR as well as FWISCBR, GOCBR has the highest prediction accuracy in the empirical test for real-world breast cancer diagnosis case. We expect that our suggest model can be applied as the computer-based expert systems or decision support systems in real-world situations. Figure 4 shows the sample screen of the Web-based expert system for helping breast cancer diagnosis using our proposed model. As you can see from Figure 4, the system suggests the predicted diagnosis results for the inputted image as well as the evidences for the prediction – the cases that are used for reference.

However, there are some limitations in this study. First of all, the size of the data set in this experiment is quite small to validate the usefulness of our proposed model. As a matter of fact, it is inevitable problem because most of medical problems use small size of samples due to the difficulty of collecting samples. Consequently, additional efforts such as 5 or 10-fold cross validation should be done in the future to mitigate the small sample problems.

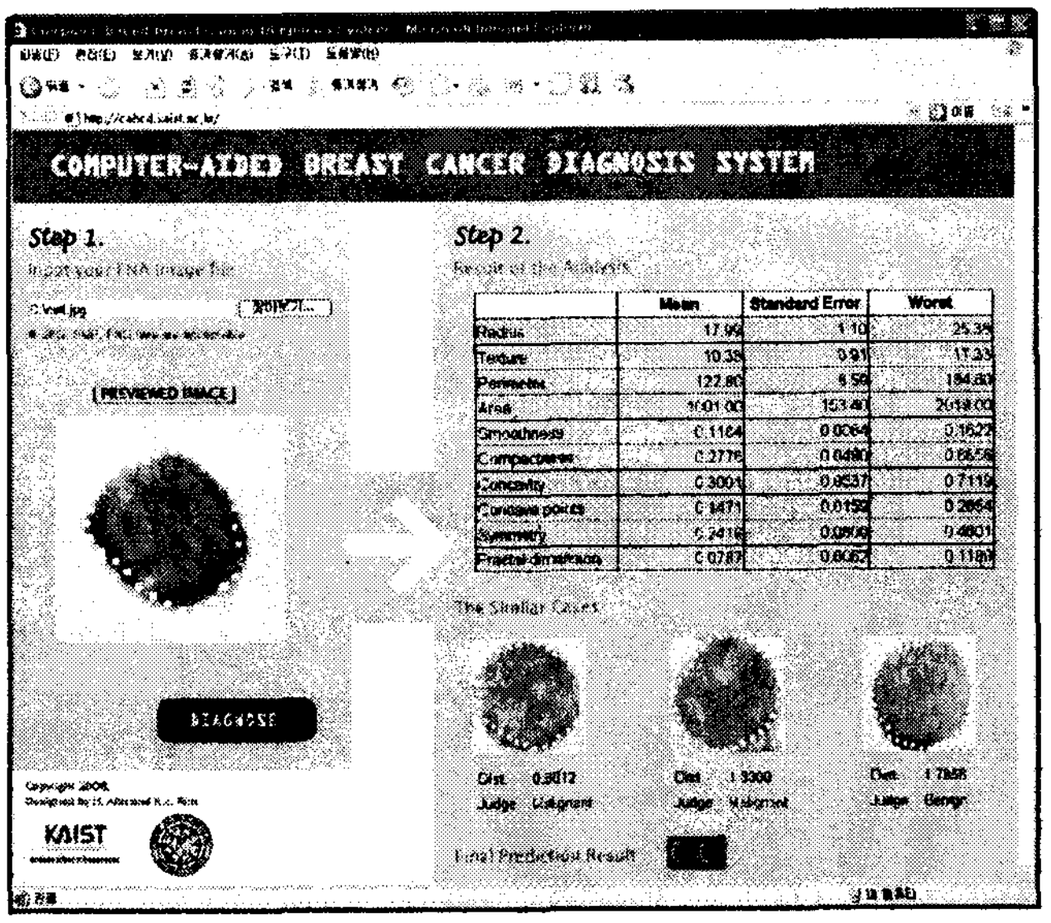


*Figure 4 – Sample screen of the Web-based expert system for helping breast cancer diagnosis*

Second, 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.

Third, 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.

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 medical domain. However, GOCBR can be applied to any other problem-solving issues in engineering, finance, and marketing domains. Thus, GOCBR should be tested and validated further in other domains in the future.

## References

[1] Aamodt, A. & Plaza, E. (1994). Case-based reasoning; Foundational issues, methodological variations, and system approaches. *AI Communications* 7(1), 39-59.

[2] Ahn, H., Kim, K.-j. & Han, I. (2003). Determining the optimal number of cases to combine in an effective case-based reasoning system using genetic algorithms. In *Proceedings of International Conference of Korea Intelligent Information Systems Society 2003 (ICKIISS*

*2003)*, Seoul, Korea, 178-184.

[3] Ahn, H., Kim, K.-j. & Han, I. (2006). Hybrid genetic algorithms and case-based reasoning systems for customer classification. *Expert Systems* 23(3), 127-144.

[4] Babu, T.R. & Murty, M.N. (2001). Comparison of genetic algorithm based prototype selection schemes. *Pattern Recognition* 34(2), 523-525.

[5] Blake, C.L. & Merz, C.J. (1998). UCI Repository of Machine Learning Database, Department of Information and Computer Science, University of California, Irvine, CA, http://www.ics.uci.edu/ mlearn/MLRepository.html.

[6] Cardie, C. (1993). Using decision trees to improve case-based learning. In *Proceedings of the 10th International Conference on Machine Learning*, 25-32.

[7] Cardie, C. & Howe, N. (1997). Improving minority class prediction using case-specific feature weights. In *Proceedings of the 14th International Conference on Machine Learning*, 57-65.

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

[9] Domingos, P. (1997). Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review* 11(1-5), 227-253.

[10] Estevez, J., Alayon, S., Moreno, L., Aguilar, R., & Sigut, J. (2002). Cytological breast fine needle aspirate images analysis with a genetic fuzzy finite state machine. In *Proceedings of the 15th IEEE Symposium on Computer-Based Medical Systems*, 21-26.

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

[12] Hart, P.E. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory* 14(3), 515-516.

[13] Huang, Y.S., Chiang, C.C., Shieh, J.W. & Grimson, E. (2002). Prototype optimization for nearest-neighbor classification. *Pattern Recognition* 35(6), 1237-1245.

[14] Humphreys, P., McIvor, R. & Chan, F. (2003). Using case-based reasoning to evaluate supplier environmental management performance. *Expert Systems with Applications* 25(2), 141-153.

[15] Jarmulak, J., Craw, S. & Rowe, R. (2000). Self-optimizing CBR Retrieval. In *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence*, 376-383.

[16] Kelly, J.D.J. & Davis, L. (1991). Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm. In *Proceedings of the 4th International Conference on Genetic Algorithms*, 377-383.

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

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

[19] Kuncheva, L.I. & Jain, L.C. (1999). Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters* 20(11-13), 1149-1156.

[20] Lee, H.Y. & Park, K.N. (1999). Methods for Determining the optimal number of cases to combine in an effective case based forecasting system. *Korean Journal of Management Research* 27(5), 1239-1252.

[21] Liao, T.W., Zhang, Z.M. & Mount, C.R. (2000). A case-based reasoning system for identifying failure mechanisms. *Engineering Applications of Artificial Intelligence* 13(2), 199-213.

[22] Lipowezky, U. (1998). Selection of the optimal prototype subset for 1-NN classification. *Pattern Recognition Letters* 19(10), 907-918.

[23] Park, Y.-J., Kim, B.-C., & Chun, S.-H. (2006). New knowledge extraction technique using probability for case-based reasoning: application to medical diagnosis. *Expert Systems* 23(1), 2-20.

[24] Rozsypal, A. & Kubat, M. (2003). Selecting representative examples and attributes by a genetic algorithm. *Intelligent Data Analysis* 7(4), 291-304.

[25] Sanchez, J.S., Pla, F. & Ferri, F.J. (1997). Prototype selection for the nearest neighbour rule through proximity graphs. *Pattern Recognition Letters* 18(6), 507-513.

[26] 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.

[27] Siedlecki, W. & Sklanski, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* 10(5), 335-347.

[28] Skalak, D.B. (1993). Using a genetic algorithm to learn prototypes for case retrieval and classification. In *Proceedings of the 1993 AAAI Workshop on Case-Based Reasoning*, 64-69.

[29] Skalak, D.B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the 11th International Conference on Machine Learning*, 293-301.

[30] Stearns, S. (1976). On selecting features for pattern classifiers. In *Proceedings of the 3rd International Conference on Pattern Recognition*, 71-75.

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

[32] Wettschereck, D., Aha, D.W. & Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review* 11(1-5), 273-314.

[33] Wilson, D.L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics* 2(3), 408-421.

[34] Yan, H. (1993). Prototype optimization for nearest neighbor classifier using a two-layer perceptron. *Pattern Recognition* 26(2), 317-324.

[35] Yin, W.J., Liu, M. & Wu, C. (2002). A genetic learning approach with case-based memory for job-shop scheduling problems. In *Proceedings of the First International Conference on Machine Learning and Cybernetics*, 1683-1687.

[36] Yu, K., Xu, X., Ester, M. & Kriegel, H.-P. (2003). Feature weighting and instance selection for collaborative filtering: an information-theoretic approach. *Knowledge and Information Systems* 5(2), 201-224.