Cost-Sensitive Case Based Reasoning using Genetic Algorithm: Application to Diagnose for Diabetes

Yoon-Joo Park[a*], Byung-Chun Kim[a]

[a]KAIST, 207-43 Cheongryangri-dong, Dongdaemoon-gu, Seoul 130-722, Korea, Tel:82-2-958-3697, Fax: 82-2-958-3604, E-mail : mirage20@kgsm.kaist.ac.kr bckim@kgsm.kaist.ac.kr

Abstract

Case Based Reasoning (CBR) has come to be considered as an appropriate technique for diagnosis, prognosis and prescription in medicine. However, conventional CBR has a limitation in that it cannot incorporate asymmetric misclassification cost. It assumes that the cost of type1 error and type2 error are the same, so it cannot be modified according to the error cost of each type. This problem provides major disincentive to apply conventional CBR to many real world cases that have different costs associated with different types of error. Medical diagnosis is an important example. In this paper we suggest the new knowledge extraction technique called Cost-Sensitive Case Based Reasoning (CSCBR) that can incorporate unequal misclassification cost. The main idea involves a dynamic adaptation of the optimal classification boundary point and the number of neighbors that minimize the total misclassification cost according to the error costs. Our technique uses a genetic algorithm (GA) for finding these two feature vectors of CSCBR. We apply this new method to diabetes datasets and compare the results with those of the cost-sensitive methods, C5.0 and CART. The results of this paper show that the proposed technique outperforms other methods and overcomes the limitation of conventional CBR.

(Keywords: Case Based Reasoning; Cost-Sensitive Learning; Genetic Algorithm; Statistics; Artificial Intelligence; Dynamic Adaptation.)

1. Introduction

Case Based Reasoning (CBR) has come to be considered as an appropriate technique for diagnosis, prognosis and prescription in medicine because the medical domain put more stress on real cases than other domains. CBR can provide knowledge obtained from real cases and is able to explain which previous cases were used for these results.

However, conventional CBR is limited in that it cannot incorporate asymmetric misclassification cost. It assumes the costs of type1 error and type2 error are equal, so it cannot produce desirable results for many real world problems when the error costs of each type influence the result. One of the most important examples where the cost of unequal misclassification error exists is the medical domain. For example, in the medical area, type1 error means a patient has a disease but the diagnosis result says he/she does not have and type2 error means a patient does not have a disease but the diagnosis result says he/she does Both of them exert a harmful influence, but for in most cases, type1 error is more dangerous than type2 error. It is because the cost of type1 error means the loss of a chance to provide medical treatment, and then the patient deteriorates, or in the worst case, the patient dies. On the other hand, the cost of type2 error means a patient just has to take an additional medical examination. For this reason, the limitation of conventional CBR provides major disincentive for application to medical problems.

In this paper we suggest the new knowledge extraction technique called Cost-Sensitive Case Based Reasoning (CSCBR) that can incorporate unequal misclassification into CBR models. The objective of CSCBR is minimizing total misclassification cost for binary class dataset that has the unequal misclassification cost. The main idea is finding the optimal cost-sensitive CBR model by adjusting the cut-off point for classifying the absence or presence of diseases, and the cut-off probability for selecting neighbors. Our technique uses a genetic algorithm (GA) to find these two feature vectors for CSCBR. To the best of our knowledge, it is the first effort to incorporate misclassification error costs to CBR. This is the extension of our previous research Statistical CBR (SCBR) that dynamically adapts the optimal number of neighbors by considering the distribution of distances between potential similar neighbors for each target case. (Park et al., 2006) SCBR overcomes the limitation of conventional CBR that

* Corresponding author. Tel.: +82-2-958-3697; Fax: +82-2-958-3604; E-mail:mirage20@kgsm.kaist.ac.kr

retrieves the fixed number of neighbors, but it still has the limitation in that it cannot incorporate asymmetric misclassification cost like other CBR models. We make up for this drawback by dynamically adjusting the cut-off point as well as the cut-off probability.

We apply this new method using diabetes dataset taken from the UCI repository on machine learning (Blake and Merz, 1998). Our reason for concentrating on the area of medical diagnosis is motivated by the fact that medical domain is the typical example that has the asymmetric misclassification error cost as well as CBR has come to be considered as an appropriate technique for medicine. We compare the results with those of the cost-sensitive methods, C5.0 and CART. The results of this paper show that the proposed technique outperforms other methods and overcomes the limitation of conventional CBR.

The rest of this paper is organized into five sections. Section 2 presents the research background. Section 3 introduces the new case extraction technique, called Cost-Sensitive Case Based Reasoning (CSCBR). Next, in section 4, a case study applied to the classification problem of diabetes diagnosis is presented. Section 5 discusses the results of the case study and evaluates its accuracy by comparing it with C5.0 and CART. Finally, concluding remarks and future research are discussed in section 6.

2. Research Background

2.1 Case Based Reasoning

Case Based Reasoning (CBR) is an approach for solving a new problem by remembering a previous similar situation and by reusing information and knowledge of that situation (Aamodt and Plaza, 1994). This concept assumes that similar problems have similar solutions, so CBR is an appropriate method for a practical domain focused on real cases rather than on rules or knowledge to solve problems. A general CBR cycle is described by the following four processes by Aamodt and Plaza (1994). :

1. RETRIEVE the most similar case or cases.
2. REUSE the information and knowledge in that case to solve the problem.
3. REVISE the proposed solution.
4. RETAIN the parts of this experience likely to be useful for future problem solving.

According to this process, CBR solves a problem by retrieving one or more previous cases, reusing them to solve the problem, revising the potential solution based on the previous cases, and retaining the new experience by incorporating it into the existing case-base (Aamodt and Plaza, 1994). One of the issues for using this conventional CBR is how many previously experienced cases to retrieve. The conventional CBR technique retrieves a fixed number of neighbors in observational space. Thus, it always selects the same number of neighbors irrespective of an optimal number of similar neighbors according to target cases. This fixed number of neighbors raises a problem when some target cases should consider more similar cases while others fewer ones. Also, a problem occurs with conventional CBR when there are too many cases equal distant from target cases. Thus, it does not guarantee optimal similar neighbors for various target cases, which leads to the weakness of lowering predictability due to deviation from desired similar neighbors. Park et al. (2006) suggest a new case extraction technique called Statistical Case Based Reasoning (SCBR) that dynamically adapts the optimal number of neighbors by considering the distribution of distances between potential similar neighbors for each target case. It selects neighbors based on similarity rather than number (Park et al., 2006). However SCBR cannot adapt the model according to the misclassification error cost, in other words, it is not a cost-sensitive method. The brief outline of SCBR can be summarized as follow.

- ***The Outline of SCBR***

Step 1. Scale data.
Step 2. Learn the distribution of distances of the learning dataset.
Step 3. Find the optimal cut-off probability from the learning dataset.
Step 4. Select neighbors within the distance threshold calculated from the obtained optimal cut-off probability for the validation dataset from Step 2 and Step 3.
Step 5. Perform CBR using selected neighbors and calculate the result.

2.2 Cost-Sensitive Methods

Cost-sensitive classification method adapts models according to the misclassification error cost in learning stage. They assume that different types of error have different cost, thus their objective is minimizing the misclassification cost instead of maximizing the classification accuracy. Many practical classification problems have different costs associated with different types of error, so recently various algorithms for cost-sensitive classification are produced. (Zhu and Wu, 2004)

Classification and Regression Tree (CART) and C5.0 are the well-known decision tree learners incorporating asymmetric misclassification cost. CART is a binary decision tree algorithm, which has exactly two branches at each internal node (Breiman et al., 1984). C5.0 is an improved version of C4.5 that can produce a cost-sensitive tree when given a cost matrix, while the previous version, C4.5 treats all misclassification error costs as equal (Quinlan, 1993; Quinlan, 1997). Nanda and Pendharkar (2001) suggest linear models based on genetic algorithm (GA) for unequal misclassification costs for bankruptcy prediction. Ting (2002) introduces an instance-weighting

method to induce cost-sensitive trees and Gama (2000) suggests a cost-sensitive iterative Bayes. Majumder (2005) suggests the Relevance Vector Machine for optical diagnosis for cancer that can handle asymmetric misclassification cost. Also, Turney (1995) introduces a new algorithm called ICET for cost-sensitive classification that considers both the costs of tests and the costs of classification errors.

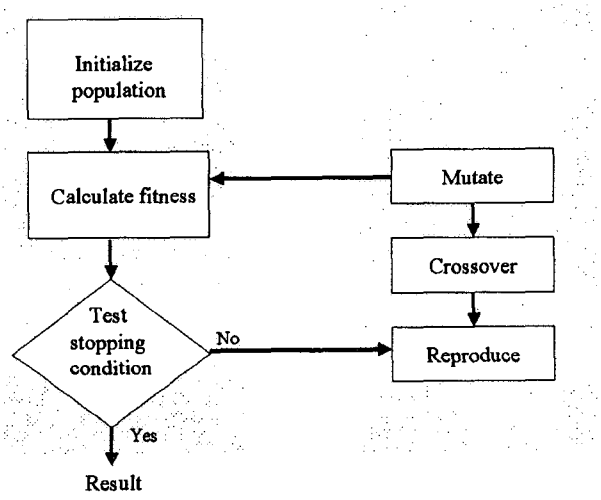
There are several methods for measuring the performance of cost-sensitive classifiers such as total misclassification cost , sensitivity , specificity and Receiver Operating Charateristic (ROC) curve (Lavrac, 1999). Total misclassification cost is a widely used method and more appropriate than accuracy for cost-sensitive classifiers (Zhu and Wu, 2004). Sensitivity is also widely used especially for the medical area because it can be viewed as a detection rate of diseases. However, sensitivity and specificity are negatively correlated. For example, if the misclassification cost of a false absence (FA) is higher than a false presence (FP) then sensitivity tends to increase and specificity decrease. The ROC curve is a two-demensional visualization of the FP rate (1-specificity, plotted on X-axis) and true presence (TP) rate (sensitivity, plotted on Y-axis). This can show that different costs make tradeoffs available for different classification threshold values. From a visual perspective, one point in the ROC curve is better than another if it is located more to the northwest, that means TP is higher and FP is lower on the ROC graph graph (Provost and Fawcett, 1997). Some research uses the area under the ROC curve (AUC) as a measurement of a classifier's performance because it can be a single figure to use as a measurement of the cost-sensitive classifiers (Bradley, 1997). But, the ROC curve and AUC cannot compare the performance of each classifier in a given error cost. The reason is that the X-axis or Y-axis of the ROC curve cannot be used as a standard axis to understand the performance in a given cost.

2.3 Genetic Algorithms

Genetic algorithms (GA) are search algorithms based on the natural selection and genetics developed by John Holland in 1975. They combined survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search (Goldberg, 1989). GA can optimize with both continuous and discrete parameters, it can deal with a large number of parameters, it can jump out of local optimum, and it does not require derivative information (Haupt, 1998). These characteristics of GA cause many researchers to use it for finding optimal values.

There are a number of variations on this algorithm, but the basic procedure of GA is composed of three operators-reproduction, crossover and mutation (Goldberg, 1989). Reproduction is a process in which individual strings representing chromosomes are copied according to their fitness function (Goldberg, 1989). Thus, the higher fitness the individual strings gets, the greater the chance it can be reproduced. Crossover means the members of the newly reproduced strings in the mating pool are mated randomly and cross over some part of each other (Goldberg, 1989). This process can diversify the individuals and gives a chance to find a better optimum. Mutation is the process that changes a certain part of strings randomly (Goldberg, 1989). For doing this, GA can jump out of local optimum, and it has a chance to converge into the global optimum. Figure 1 shows the process of GA.

There are some previous studies that hybridize GA and CBR. Kim (2004) uses GA for selecting a relevant feature subset and optimizing feature weights of CBR for financial forecasting. Golobardes (2002) incorporates CBR and GA to diagnose breast cancer. Chiu (2002) presents a GA based approach to enhance the case-matching process. Likewise there is other research that introduces GA to find the optimal value for CBR (Kim and Han, 2001; Fu and Shen, 2004; Hsua et al., 2004; Chiu et al., 2004).



[Figure 1] Procedure of genetic algorithm (GA)

3. Cost-Sensitive Case Based Reasoning

Unequal misclassification costs are common in real-world situations. Especially for medical diagnosis, asymmetric misclassification costs have to be considered as an important factor. In this article, we suggest a new case based reasoning method called Cost-Sensitive Case Based Reasoning (CSCBR) that can incorporate unequal misclassification into CBR models. The main idea of CSCBR is finding the optimal cut-off point for classifying the absence or presence of diseases and the cut-off probability for selecting neighbors using a genetic algorithm (GA). The cut-off point is the boundary point for classifying the result. The cut-off probability means the probability that a case can be a neighbor based on proximity, so certain cases can be neighbors of the target case if the distance between the cases and the target case is

less than the cut-off probability. The concept of CSCBR is finding the optimal value of these two feature vectors of CBR that can minimize cost.

In this section, we explain the overall procedure of CSCBR. In the first step, it performs exploratory data analysis. In the second step, it transforms data by standardization to eliminate the effects of units or measurement, and it determines the learning phase and the validation phase. In the third step, CSCBR finds the optimal cut-off probability and the cut-off probability using GA in the learning phase. In the fourth step, it selects the neighboring cases in the learning dataset that satisfies the optimal cut-off probability criterion and calculates results using them for targeting cases. After that it classifies the results using the optimal cut-off point obtained in the previous step. In the fifth step, it detransforms the variables and finally evaluates performance. The whole procedure of CSCBR is presented in Figure 2. The core procedures of CSCBR are step3 and step4 in Figure 2, so we provide more detail on them in the rest of this section.

3.1 Classify the diagnosis result

In this section, we will explain how to classify the diagnosis result by adapting the cut-off point. Usually the cut-off point of conventional CBR lies exactly the halfway between the two result points because it assumes the same error cost regardless of the types of error, but CSCBR adapts the cut-off point according to the misclassification error cost. For example, if we represent absence of disease as 0 and the presence of disease as 1 then the cut-off point, p_{opt} defines the boundary classification point between 0 and 1. Then, the position of p_{opt} is adapted as the type1 error cost and type2 error cost change. Intuitively, we expect the cut-off point, p_{opt} to move toward 0 if the type1 error cost increases, and we expect p_{opt} to move toward 1 if the type2 error cost decreases. Because the higher type1 error cost means it can cause a more harmful effect with a false absence, CSCBR will put more stress on preventing this kind of misclassification by providing more diagnoses that disease is present. CSCBR is optimized to find the cut-off point, p_{opt} and cut-off probability, α to minimize the total misclassification cost using GA introduced in section 3.3. Figure 3 presents the misclassification cost matrix and Figure 4 shows the case where the cut-off point, p_{opt} is skewed to 0 to prevent type1 error.

		Diagnosis result	
Real result		0 (absence)	1 (presence)
	0 (absence)		Type2 error
	1 (presence)	Type1 error	

(Type1 error: Patient has the disease but the diagnosis result says it does not have.
Type2 error: Patient does not have the disease but the diagnosis result says it has.)

[Figure 3] Misclassification cost matrix

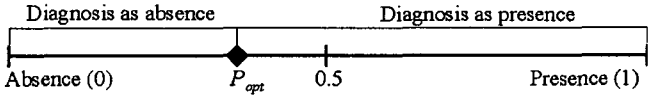
1. Perform exploratory data analysis (EDA); identify overall patterns and outliers.
2. Transform data for comparability:
   a. Determine the learning phase and validation phase.
   b. Eliminate effects of units (of measurement) by subtracting mean and dividing by standard deviation if attributes are real type: $V_{ij} \rightarrow ZV_{ij} \equiv Z_{ij}$
3. Find the optimal cut-off point between the binary diagnosis results and the cut-off probability for the learning dataset using genetic algorithm by section 3.3.
   a. Select several pairs of cut-off point, p_{opt} and cut-off probability, α_{opt} randomly.
   b. Calculate the performances at those points.
   c. Reproduce more number of the points that produce the higher performances.
   d. Cross over two points randomly selected.
   e. Mutate part of points randomly selected.
   f. Repeat step 3.a to step 3.e until the all of the points converge into the specific position.
4. Perform CSCBR method using the optimal cut-off point, p_{opt} and cut-off probability, α_{opt} founded in step3 for the validation dataset.
   a. Begin with target case $x(t_c)$ in the validation phase.
   b. Seek the neighboring cases $x(t_i)$ in the learning phase that is on the inside of the distance threshold, $D_{threshold}$, according to the distance function:
      - $d_i \equiv d[x(t_i), x(t_c)]$
      - $d_i < D_{threshold}$

      ($D_{threshold} = e^{\mu + z_\alpha \times \sigma}$

      μ: Mean of distances σ: Standard deviation of distances)
   c. Compute the sum of weights: $d_{TOT} = \sum_{i=1}^{J} d_i$
   d. Determine the relative weight of i^{th} neighbor: $w_i = \frac{1}{J-1}\left[1 - \frac{d_i}{d_{TOT}}\right]$
   e. Find the output attribute $o(t_i)$ of each case $x(t_i)$ in the set of neighbors.
   f. Calculate the result for the target case $x(t_c)$ as the weighted sum of output attributes: $\hat{o}(t_c) = \sum_{i=1}^{J} w_i o(t_i)$
   g. Decide the diagnosis result between the binary results 0 and 1 by the boundary line that the optimal cut-off point, p_{opt} defines:
      - If $\hat{o}(t_c) > p_{opt}$ then $\hat{r}(t_c) = 1$

        else $\hat{r}(t_c) = 0$
5. Detransform the variables.
6. Evaluate performance.

[Figure 2] The process of the proposed CSCBR



[Figure 4] The Cut-off point, p_{opt} for classifying the diagnosis result

3.2 Finding neighbors using distribution of distances

One issue for using conventional CBR is how many previously experienced cases to retrieve, since it can strongly influence the performance of CBR. Generally many CBR models use a fixed number of neighbors without considering an optimal number for each target case, so it does not guarantee optimal similar neighbors for various target cases. Park el al. (2006) suggested case extraction method called SCBR that retrieves neighbors based on the probabilistic similarity of rather than number. CSCBR retrieves neighbors based on this method.

First, we analyze the distribution of distances between cases. For this, we calculate every distance between cases using Euclidean distance measurement. Consider distances between cases as a random variable D then the range is $0<D<\infty$.

$$D_{ij} = \sqrt{(X_{i1} - X_{j1})^2 + (X_{i2} - X_{j2})^2 \ldots + (X_{ik} - X_{jk})^2}$$

(D_{ij}: Distance between case i and case j.
k : The number of variables
X_{ik} : K $_{th}$ value of variable X for case i
X_{jk} : K $_{th}$ value of variable X for case j)

The distribution of the distances is not normally distributed in many cases, so we transform the original distance data by log transformation using a natural logarithm. Log transformation is one of the most prevalent and effective methods for transforming positive or skewed random variables into normal distribution. We transform D by natural logarithm and denote this as random variable Y and assume that Y is normally distributed.

$Y_{ij} = \ln(D_{ij})$ (D_{ij} : Distance between case i and case j
Y_{ij} : Log transformed distance between case i and case j)

$Y \sim N(\mu, \sigma^2)$
(Y: Random variable of log transformed distances
μ : Average distance of log transformed distances
σ^2: Variance of log transformed distances)

Second, determine the distance threshold to be selected as one of the neighbors for a certain cut-off probability, α, that is given. This cut-off probability means the probability that the distance D_{ij} between case i and j is less than the given threshold. Thus, if α is 0.05, then the proportion of cases when the distance is less than the threshold is 5%. The method to find that optimal cut-off probability, α_{opt} is introduced later in section 3.3.

$P[Y < \ln(D_{threshold})]$ ($D_{threshold}$: Distance threshold of neighbor
$= P\left[\frac{Y-\mu}{\sigma} < \frac{\ln(D_{threshold})-\mu}{\sigma}\right]$ α: Cut-off probability)
$= P\left[Z < \frac{\ln(D_{threshold})-\mu}{\sigma} (= Z_\alpha)\right]$

Thus, the region of Y for selecting neighbors is $Y < \mu + z_\alpha \times \sigma$, and the region of D is $D < exp(\mu + z_\alpha \times \sigma)$. However, we don't know the value of μ and σ, so we estimate these values with sample mean $\overline{Y}$ and sample standard deviation S. Now, the following region is constructed.

$$Y(= \ln(D)) < \overline{Y} + z_\alpha \times S$$

So, $D < \exp(\overline{Y} + z_\alpha \times S)$

Conclusively, the distance threshold between potential neighbors and the target case to be a neighbor is $e^{\overline{Y}+z_\alpha \times S}$.

Third, select neighbors from the training dataset if they have smaller distances than the distance threshold, $e^{\overline{Y}+z_\alpha \times S}$.

3.3 Finding the optimal cut-off point and cut-off probability for minimizing cost

To find the optimal cut-off point and cut-off probability, we use a genetic algorithm (GA) explained in section2.3. So, in this section, we explain how to apply GA to find the optimal cut-off point and cut-off probability for minimizing cost. First, the algorithm randomly selects several pairs of cut-off points and cut-off probabilities. Second, it calculates the total misclassification cost for these pairs. Third, it reproduces the same number of pairs. In this stage, the values that produce lower total misclassification cost are reproduced more than others. Fourth, it crosses over two pairs that are randomly selected in the probability P_{cross} that is given. Fifth, it mutates some part of the values that are represented as binary digits. Finally, it repeats the previous procedure until all of the values converge into the specific position or the repetition is over. Figure 5 presents the GA applied for searching for the optimal cut-off point and cut-off probability for minimizing the cost.

4. Case Study

4.1. The data

We execute the case study using diabetes dataset obtained from the UCI repository (Blake and Merz, 1998). The dataset was collected by the National Institute of Diabetes and donated by V. Sigillito. The dataset originally contained 768 cases and 9 attributes, but we use 760 cases to construct 10 subsets of equal size. It consists of 2 classes where 492 cases show the presence of diabetes and 268 cases when it is absent. We use the first 8 attributes to diagnose diabetes and compare the results with the final attribute. Table 1 shows the distribution of the dependent variable.

Category	Frequency	Percentage
0 (Absence of disease)	492	64.74 %
1 (Presence of disease)	268	35.26 %
Total	760	100 %

[Table 1] Distribution of the dependent variable (Diabetes)

1. Randomly select several pairs of cut-off point, X_i and cut-off probability, Y_i :
   Initial chromosome set = $[(X_1,Y_1), (X_2,Y_2), (X_3Y_3), \ldots, (X_k,Y_k)]$
   $(i=1..K, \quad L_x<X_i<U_x \quad L_y<Y_i<U_y)$
2. Calculate the fitness probability for each chromosome.
   a. Calculate the costs: $C_i = Cost\,[(X_i, Y_i)] \quad (i=1..K)$
   b. Compute the sum of costs: $C_{tot} = \sum_{i=1}^{K} C_i$
   c. Calculate the fitness: $F_i = 1 - \frac{C_i}{C_{tot}} \quad (i=1..K)$
   d. Compute the sum of fitness: $F_{tot} = \sum_{i=1}^{K} F_i$
   e. Calculate the probabilities: $P_i = \frac{F_i}{F_{tot}} \quad (i=1..K)$
3. Reproduce more number of chromosomes that have higher probability, P_i.
   a. Compute cumulative probability of each chromosome: $U_i = \sum_{j=1}^{i} U_j$
   b. Produce a random number R_1, that $0<R_1<1$.
   c. Reproduce i_{th} chromosome (X_i, Y_i) if $U_{i-1} < R < U_i$.
   d. Repeat from 3.a to 3.c until K numbers of chromosomes are reproduced.
4. Cross over chromosomes in the probability P_{cross} that is given.
   a. Randomly select two chromosomes (X_i, Y_i) and (X_j, Y_j).
   b. Compute length, l of chromosome to transform into binary code.
      $l_x = log_2(U_x - L_x) * 10^d + 1$
      $l_y = log_2(U_y - L_y) * 10^d + 1$
      $l = l_x + l_y$ (l_x: binary code length of X_i , l_y: binary code length of Y_i)
   c. Transform each chromosome into binary code:
      $(X_i, Y_i) \rightarrow B_{i1}B_{i2}B_{i3}\ldots B_{il}$
      $(X_j, Y_j) \rightarrow B_{j1}B_{j2}B_{j3}\ldots B_{jl}$
   d. Produce a random number R_2, that $1<R_2<l$ and cross over after R_2 position.
      R_2
      ↓ Cross over
      $B_{i1}B_{i2}B_{i3}\mathbf{B_{i4}}\,\mathbf{B_{i5}}\ldots\mathbf{B_{i(l-1)}}\mathbf{B_{il}} \Rightarrow B_{i1}B_{i2}B_{i3}\mathbf{B_{j4}}\,\mathbf{B_{j5}}\ldots\mathbf{B_{j(l-1)}}\mathbf{B_{jl}}$
      $B_{j1}B_{j2}B_{j3}\mathbf{B_{j4}}\,\mathbf{B_{j5}}\ldots\mathbf{B_{j(l-1)}}\mathbf{B_{jl}} \quad B_{j1}B_{j2}B_{j3}\mathbf{B_{i4}}\,\mathbf{B_{i5}}\ldots\mathbf{B_{i(l-1)}}\mathbf{B_{il}}$
5. Mutate chromosome in the probability P_{mutate} that is given.
   a. Produce a random number R_3, $0<R<1$.
   b. Mutate chromosome if $R<P_{mutate}$.
   c. Repeat step 5.a and step5.b until the end of the chromosomes.

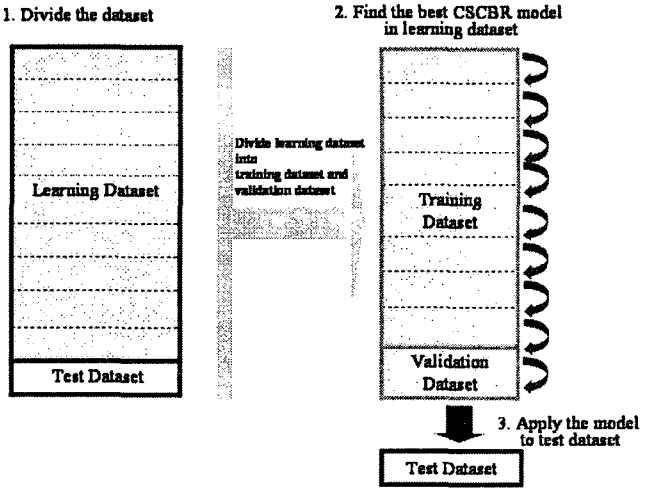
Repeat step 2 to step 5 until the all of the chromosomes converge into the specific position.

[Figure 5] Genetic algorithm for searching the optimal cut-off point and cut-off probability

4.2. Model construction – k-fold cross-evaluation

The classification model is trained and tested 10 times. In 10-fold cross-validation the entire dataset is divided into 10 mutually exclusive subsets with the same class distribution. Each fold is used once to test the performance of the classifier that is generated from the combined data of the remaining 9 folds. Specifically, we use the following three-step 10-fold cross-validation procedure to estimate the performance:

1. Divide the dataset into learning dataset and test dataset.
2. Find the model that best performs in learning dataset by 9-fold cross-validation.
3. Apply the model to each validation dataset by 10-fold cross-validation.



[Figure 6] Graphical design of the cross validation experiment

4.3 Performance Evaluation criterion

We use total misclassification cost, sensitivity and specificity to measure the performances of the models. There are several other methods that we introduced in section 2.1, but we use total misclassification cost as the primary measurement because it is an appropriate method to evaluate performance for cost-sensitive learning in a given error cost condition. Total misclassification cost measures overall cost of incorrectly classified cases in given error cost. Sensitivity measures the fraction of presence cases that are classified as presence, and specificity measures the fraction of absence cases classified as absence (Lavrac, 1999).

Total Misclassification Cost
$= Number_{FP} * Cost_{FP} + Number_{FA} * Cost_{FA}$

$$Sensitivity = \frac{Number_{TP}}{Number_{TP} + Number_{FA}} \quad (= TP\text{-}rate)$$

$$Specificity = \frac{Number_{TA}}{Number_{TA} + Number_{FP}} \quad (= 1 - (FP\text{-}rate))$$

($Cost_{FP}$: Misclassification error cost of false present (=Type2 error cost)
$Cost_{FA}$: Misclassification error cost of false absent (=Type1 error cost)
$Number_{FP}$: The Number of misclassified cases of false present
$Number_{FA}$: The Number of misclassified cases of false absent
$Number_{TP}$: The Number of misclassified cases of true present
$Number_{TA}$: The Number of misclassified cases of true absent
)

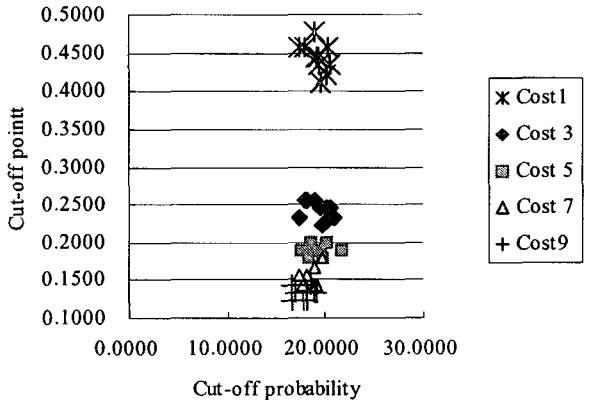
- True absent (TA): Patient who does not have the disease and the diagnosis result is correct.
- True present (TP): Patient who has the disease and the diagnosis result is correct.

• False absent (FA): Patient who has the disease but the diagnosis result is incorrect.
• False present (FP): Patient doesn't have the disease but the diagnosis is incorrect.

5. Results of Study

In this section we show the result of the case study experiments introduced in section 4. We compare the performance of CSCBR with other cost-sensitive knowledge discovery methods, C5.0 and CART by changing $Cost_{FA}$ five times – 1,3,5,7,9- against $Cost_{FP}$ fixed as 1. We implement Cost-Sensitive CBR by JAVA and other data mining techniques such as C5.0 and CART by commercial application, *Clementine8.5*.

To perform CSCBR, we find the optimal cut-off probability and cut-off point for targeting cases using a GA. We set the lower bound of cut-off probability as 1% and the upper bound as 30%, and we set the lower bound of the cut-off point as 0, which means absence of disease, and the upper bound as 1, which means presence. We use 50 to 100 chromosomes in the population and cross them over in the probability P_{cross}, 0.8, and the chromosomes mutate in the probability P_{mutate}, 0.03. We use 10-fold cross validation for this experiment. The selected optimal cut-off probability and cut-off point for each fold is different. We experiment 10-fold cross validation for every dataset by changing $Cost_{FA}$ to 1,3,5,7,9 when $Cost_{FP}$ is 1 . Table 2 shows the results of optimal cut-off probability and the optimal cut-off point searched using GA for the diabetes dataset and Figure 7 presents these graphically. After finding the optimal cut-off probability and the optimal cut-off point for the data set, CSCBR performs using these values. Tale 3 shows the detailed performances of CSCBR using these optimized values in Table 2 for the diabetes dataset. As we expected in section 3.1, the cut-off point is likely to moves toward 0 as $Cost_{FA}$ increases. We can understand this situation because if $Cost_{FA}$ increases, CSCBR would be adapted to classify more cases with the presence of disease to prevent false absence.



[Figure 7] The optimal cut-off probability and cut-off point (Diabetes)

$Cost_{FA}$/ $Cost_{FP}$	1		3		5		7		9	
Fold No	Cut.Pr.	Cut.Po.	Cut.Pr.	Cut.Po.	Cut.Pr.	Cut.Po	Cut.Pr.	Cut.Po.	Cut.Pr.	Cut.Po.
1	20.33	0.46	20.22	0.24	21.67	0.19	17.89	0.14	18.22	0.13
2	19.33	0.43	20.44	0.24	18.00	0.19	18.89	0.17	16.89	0.13
3	20.44	0.43	19.00	0.26	18.33	0.18	18.44	0.14	18.22	0.13
4	19.00	0.48	17.33	0.23	17.56	0.19	18.22	0.16	19.22	0.13
5	19.11	0.44	20.44	0.23	20.11	0.20	18.00	0.14	16.56	0.12
6	20.11	0.42	19.78	0.22	19.11	0.19	19.00	0.14	18.89	0.13
7	19.44	0.41	19.44	0.24	19.78	0.18	17.67	0.14	17.78	0.12
8	17.89	0.46	17.89	0.26	18.56	0.20	17.33	0.16	16.67	0.14
9	19.33	0.44	20.89	0.23	19.67	0.18	19.11	0.14	18.11	0.12
10	17.44	0.46	18.22	0.26	19.44	0.19	19.78	0.18	18.56	0.14
Aver.	19.24	0.44	19.37	0.24	19.22	0.19	18.43	0.15	17.91	0.13

[Table 2] Optimal cut-off probability and cut-off point of CSCBR with 10-fold cross validation (Diabetes)

The overall results of C5.0, CART and CSCBR are presented in Table 4. Total misclassification cost of CSCBR is the lowest among all methods for diabetes dataset. Total misclassification costs of C5.0, CART and CSCBR for diabetes dataset are showed in Figure 8. We can see that the total misclassification costs do not increase in arithmetic progression as the $Cost_{FA}$ increase arithmetically thus those methods are cost-sensitive.

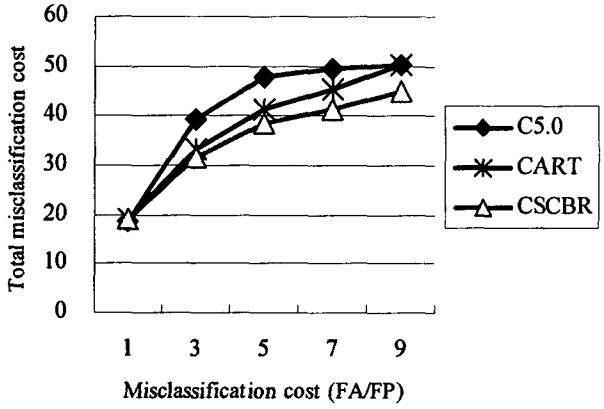
We perform a paired t-test to verify that the results are statistically significant. The null hypothesis is that total misclassification cost of CSCBR is not less than other methods. The result indicates that total misclassification cost of CSCBR is significantly less than the cost of C5.0 and CART at the 95% confidence interval for diabetes dataset. (see Table 5)

$Cost_{FA}$/ $Cost_{FP}$	1			3			5			7			9		
Fold No.	Cost	Sen.	Spe.	Cost	Sen.	Spe.	Cost	Sen.	Spe.	Cost	Sen.	Spe.	Cost	Sen.	Spe.
1	22	0.30	0.94	34	0.78	0.67	38	0.89	0.53	22	0.30	0.94	34	0.78	0.67
2	17	0.59	0.88	25	0.93	0.61	32	0.93	0.55	17	0.59	0.88	25	0.93	0.61
3	21	0.56	0.82	39	0.78	0.57	45	0.89	0.39	21	0.56	0.82	39	0.78	0.57
4	21	0.48	0.86	33	0.81	0.63	43	0.85	0.53	21	0.48	0.86	33	0.81	0.63
5	23	0.41	0.86	28	0.85	0.67	41	0.85	0.57	23	0.41	0.86	28	0.85	0.67
6	19	0.63	0.82	29	0.85	0.65	42	0.85	0.55	19	0.63	0.82	29	0.85	0.65
7	11	0.81	0.88	27	0.93	0.57	28	1.00	0.43	11	0.81	0.88	27	0.93	0.57
8	21	0.44	0.88	22	0.93	0.67	21	1.00	0.57	21	0.44	0.88	22	0.93	0.67
9	16	0.64	0.86	35	0.88	0.49	38	0.96	0.35	16	0.64	0.86	35	0.88	0.49
10	21	0.56	0.82	44	0.74	0.53	57	0.78	0.45	21	0.56	0.82	44	0.74	0.53
Aver.	19.2	0.54	0.86	31.6	0.85	0.61	38.5	0.90	0.49	19.2	0.54	0.86	31.6	0.85	0.61

[Table 3] Performances of CSCBR with 10-fold cross validation (Diabetes)

Methods	$Cost_{FA}/Cost_{FP}$	1	3	5	7	9
C5.0	Tot. Cost	18.800	39.300	47.900	49.600	50.400
	Sensitivity	0.6190	0.7476	0.8142	0.8698	0.8920
	Specificity	0.8248	0.6172	0.5355	0.4906	0.4941
CART	Tot. Cost	18.900	33.300	41.300	45.300	50.200
	Sensitivity	0.6917	0.7615	0.8396	0.8729	0.8951
	Specificity	0.8513	0.7134	0.5975	0.5628	0.4914
CSCBR	Tot. Cost	19.200	31.600	38.500	41.200	44.900
	Sensitivity	0.5418	0.8473	0.8997	0.9404	0.9553
	Specificity	0.8597	0.6082	0.4924	0.3910	0.3074

[Table 4] Average results according to misclassification cost (Diabetes)



[Figure 8] Total misclassification cost (Diabetes)

P-value	H_0: C5-CSCBR<=0	H_0: CART-CSCBR<=0
Diabetes	*0.0039*	*0.0092*

[Table 5] Overview of the pairwised t-test result

6. Concluding Remarks and Future Work

We have proposed a new CBR method called cost-sensitive case based reasoning (CSCBR) that can incorporate unequal misclassification costs into CBR and optimize the number of neighbors dynamically. Our model adapts to minimize total misclassification cost. To the best of our knowledge, it is the first effort to apply misclassification error costs to CBR. It is meaningful not only for introducing the concept of cost-sensitive learning to CBR, but also for encouraging the use of CBR in the medical area.

We apply the suggested method for diabetes dataset to verify the effectiveness of this method. The results are compared with those of C5.0 and CART. Our finding shows that the total misclassification cost of CSCBR is smaller than other cost-sensitive methods for diabetes dataset. Also the paired t-test results indicate that the total misclassification cost of CSCBR is significantly less than C5.0 and CART.

Our future work extends these experiments to many other datasets to verify that CSCBR is an effective method generally and also we will extend it for multi-classification CSCBR can be applied only for binary classification problems, thus we will try to extend CSCBR for classifying more than three groups.

References

Aamodt, A., E. Plaza (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches, *AI communications: the European journal on artificial intelligence*, 7(1), 39-59.

Blake, C.L., C.J. Merz (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).

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees. Wadsworth and Brooks*, Monterey, Ca.

Bradley, A.P. (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognition*, 30(7), 1145-1159.

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

Chiu, C., N.-H. Chiu, C.-I. Hsu (2004) Intelligent aircraft maintenance support system using genetic algorithms and case-based reasoning, *International journal of advanced manufacturing technology*, 24(5/6), 440-446.

Chun, S.-H, Y.-J. Park (2005) Dynamic adaptive ensemble case-based reasoning: application to stock market prediction, *Expert system with application*, 28(3), 435-443.

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

Gama, J. (2000) A cost-sensitive iterative Bayes, *In seventeenth international conference on machine learning, Workshop on cost-sensitive learning.*

Goldberg, D.E. (1989) Genetic algorithms in search, optimization, and machine learning, Addison-Wesley Pub. Co.

Golobardes, E., X. Llora, M. Salamo, J. Marti (2002) Computer aided diagnosis with case-based reasoning and genetic algorithms, *knowledge based systems*, 15(1/2),

Haupt, R.L., S.E. (1998) Practical genetic algorithms, Wiley.

Hsua, C.-I, C. Chiub, P.-L. Hsuc (2004) Predicting information systems outsourcing success using a hierarchical design of case-based reasoning, *Expert systems with applications*, 26, 435–441.

Kim, K.-J. (2004) Toward Global Optimization of Case-Based Reasoning Systems for Financial Forecasting, *Applied intelligence*, 21(3), 239-249.

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

Lavrac, N. (1999) Selected techniques for data mining in medicine, *Artificial intelligence in medicine,* 16(1), 3-23.

Majumder, S. K., N. Ghosh, P.K. Gupta (2005) Relevance vector machine for optical diagnosis of cancer, *Lasers in surgery and medicine*, 36(4), 323-333.

Nanda, S. and P. Pendharkar (2001) Linear Models for Minimizing Misclassification Costs in Bankruptcy prediction, *International journal of intelligent systems in accounting, Finance & management*, 10(3), 155-168.

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

Provost, F., T. Fawcett (1997) Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions, *Knowledge discovery and data mining*, 43-48.

Quinlan, J.R. (1993) C4.5: program for machine for machine learning, *Morgan Kaufmann.*

Quinlan, J.R., "C5", http://rulequest.com, 1997.

Ting, K. M., An instance-weighting method to induce cost-sensitive trees (2002), *IEEE transactions on knowledge and data engineering*, 14(3), 659-665.

Turney, P. (1995) Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm, *Journal of artificial intelligence research 2*, 369-409.

Zhu, X., X.Wu (2004) Cost-guided class noise handling for effective cost-sensitive learning, *Proceedings of the fourth IEEE international conference on data mining,*