Asia Pacific Journal of Information Systems
Vol. 25 No. 4 (DECEMBER 2015), 626-641

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
http://dx.doi.org/10.14329/apjis.2015.25.4.626

Improving Real-Time Efficiency of Case Retrieving Process for Case-Based Reasoning

Yoon-Joo Park*

Assistant Professor, Seoul National University of Science and Technology, Korea

ABSTRACT

Conventional case-based reasoning (CBR) does not perform efficiently for high-volume datasets because of case retrieval time. To overcome this problem, previous research suggested clustering a case base into several small groups and retrieving neighbors within a corresponding group to a target case. However, this approach generally produces less accurate predictive performance than the conventional CBR. This paper proposes a new case-based reasoning method called the clustering-merging CBR (CM-CBR). The CM-CBR method dynamically indexes a search pool to retrieve neighbors considering the distance between a target case and the centroid of a corresponding cluster. This method is applied to three real-life medical datasets. Results show that the proposed CM-CBR method produces similar or better predictive performance than the conventional CBR and clustering-CBR methods in numerous cases with significantly less computational cost.

Keywords: Case-Based Reasoning, Case-Retrieval, Dynamic Clustering, Real-Time Computational Cost, Medical Diagnosis

Ⅰ. Introduction

Case-based reasoning is a memory-based method which solves a new problem by retrieving previous similar cases, so called neighbors, from a case-base. Thus, the more data that has been accumulated in a case-base, the more time it takes to retrieve neighbors, which leads to prolonged prediction time in proportion to the size of a case-base (Aamodt and Plaza, 1994; Porter et al., 1990). This is a major limitation of the conventional CBR method when applying it to many real-life, high volume, or rapidly growing datasets.

In order to overcome this problem, some previous research suggests applying a clustering technique when using a CBR method (Hong and Liou, 2008; Khan et al., 2008; Kim and Han, 2001; Li et al., 2006; Park, 2013; Qiang and King, 2001). For example, they suggest clustering a case-base into several small groups off-line. After that, when a new target case comes in, the group to which the target case is involved with is determined, and a CBR method is

This study was supported by the Research Program funded by the Seoul National University of Science and Technology.

*Corresponding Author. E-mail: yjpark@seoultech.ac.kr Tel: 8229706438 performed only within the corresponding group. We call this method the clustering-CBR (*C-CBR*) method. The *C-CBR* method works well in terms of reducing real-time computational cost, since it searches neighbors only within a corresponding group instead of a whole case-base. However, it often retrieves less proximate neighbors than the conventional CBR (CBR) method, which often results in lower predictive performance. This problem is discussed in more detail in Section 3.

This paper suggests a new case-based reasoning method called the Clustering-Merging CBR (*CM-CBR*) method. The *CM-CBR* method dynamically expands a searching pool to the other adjacent clusters for retrieving more similar neighbors than the basic *C-CBR*. In other words, the *CM-CBR* method retrieves neighbors only in a corresponding cluster like the basic *C-CBR* method when a target is placed in the center of it, however, if a target case is placed in a boundary area then the *CM-CBR* method expands the searching pool to the other adjacent clusters.

The suggested *CM-CBR* method was applied to three real-life medical sets of data and the experimental results were compared with those of the CBR and the *C-CBR* methods. The results show that the suggested *CM-CBR* method produces similar or better predictive performance than the conventional CBR and the clustering-CBR(*C-CBR*) methods in many cases with significantly less computational cost.

The rest of this paper is organized into four sections. Section 2 presents the related research. Section 3 indicates the limitations of the basic clustering CBR(*C-CBR*) method and suggests a new clustering CBR method called the Clustering-Merging CBR (*CM-CBR*) method. In Section 4, the experimental results of the *CM-CBR* method are presented comparing the conventional CBR and *C-CBR* methods. Finally, concluding remarks and areas for future research are discussed in Section 5.

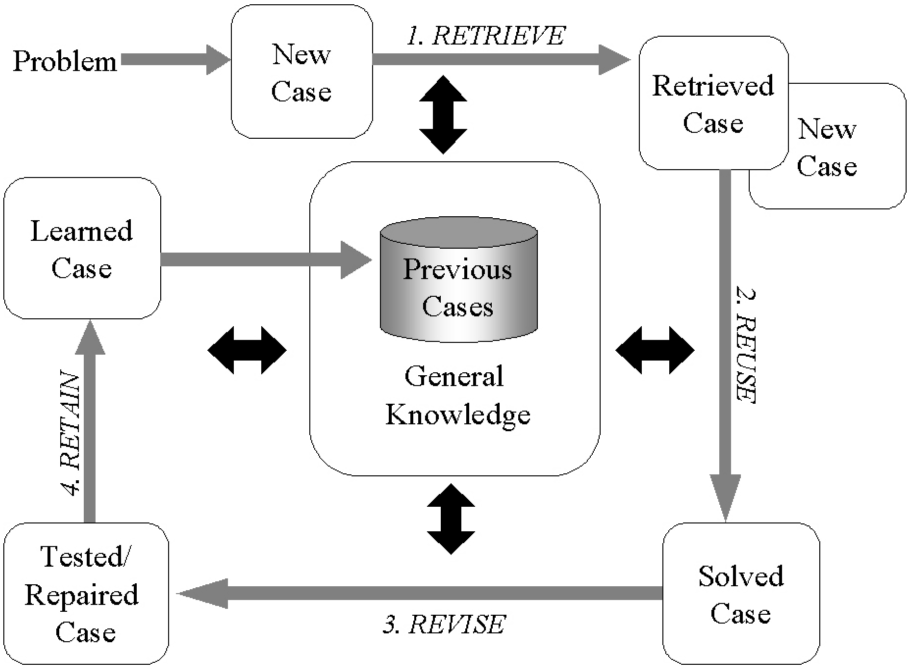
Ⅱ. Related Research

Case Based Reasoning (*CBR*) is an approach for solving a new problem by remembering a previous similar situation and by reusing information and knowledge from 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 (Porter et al., 1990). A general CBR cycle is described by the following four processes and graphically presented in <Figure 1>:

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.

However, since CBR solves a new problem by retrieving previous similar cases by comparing the target case with all other cases in a case-base; the more data that has been accumulated in a case-base, the more time it takes to retrieve neighbors. This scalability problem causes the deterioration of real-time computational performance of CBR accord-



<Figure 1> CBR Cycle Developed by Aamodt and Plaza (1994)

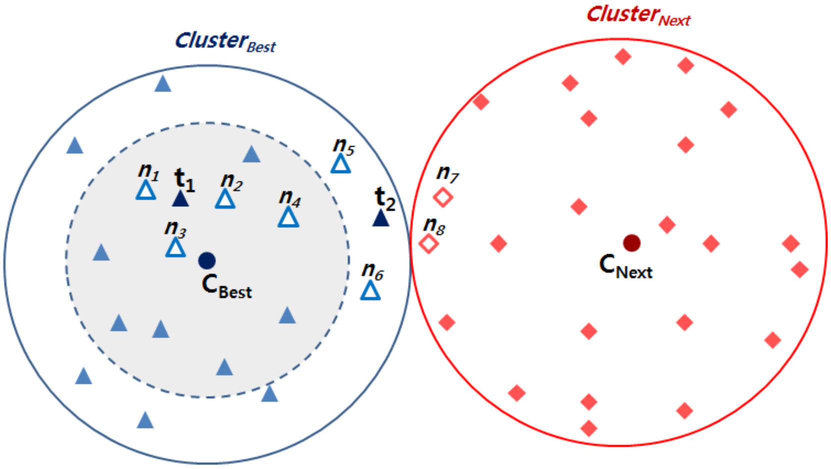
ing to the amount of a dataset and becomes the major limitation of CBR in terms of applying it to high volume or rapidly growing datasets.

In order to overcome this scalability problem, some researchers apply clustering techniques to CBR. Khan et al (2008) propose to cluster a case-base to reduce a search-space when considering small subset of cases during case retrieval, thus reducing the real-time computational costs of CBR. Qiang and Jing (2001) also use clustering algorithm in their suggested interactive case-based reasoning system called *CaseAdvisor* to compress a large case-base into several small ones. Hong and Liou (2008) apply clustering techniques for feature selection in the case retrieval process to improve efficiency of the large-scale CBR. Kim and Han (2001) apply clustering techniques for case-indexing assuming that a case-base is already clustered into some distinct subgroups. However, this previous research does not show how different types of clustering models affect the performance of clustered CBR, nor what the adequate clustering model for given data is. In our previous work related to this research, we also use clustering ideas to reduce case retrieval time (Park, 2013). However, in this method, every previous case as well as a target case should be determined whether they are placed in centered area or boundaries in advance, which greatly increases off-line computational time. Also, this method cannot expand a search space of the clustered CBR to more than two clusters. The concise version of this paper was presented in International Conference on Informatics, Management and Technology in Healthcare and published in ICIMTH 2014 proceedings (Park, 2014).

Ⅲ. The Issue of the Clustering Case-Based Reasoning (C-CBR) Methods

The *C-CBR* method works well in terms of reducing real-time computational costs of the conventional *CBR* method by reducing a search space. However, it has an issue with predictive performance. Since, the *C-CBR* method searches neighbors within a corresponding group, it usually retrieves less proximate neighbors for a target case than the conventional *CBR* method. Thus, predicting results using these less proximate neighbors often produces less accurate predictive performance than the *CBR*. Conclusively, there is a trade-off between computational cost and predictive performance when using the *C-CBR* method. Let us assume that the *C-CBR* method solves the current problem by retrieving three previous neighbors. If a target case t_1 is placed near the centroid of the corresponding cluster $Cluster_{Best}$, then the neighboring cases n_1, n_2, n_3 are the best choice for t_1 in terms of similarities. However, if a target case is placed relatively far from the centroid of a corresponding cluster, such as the target case t_2 in the <Figure 2>, then the neighboring cases of t_2 becomes n_4, n_5, n_6. This is because the conventional *C-CBR* method only finds neighbors within the corresponding cluster, $Cluster_{Best}$, even though there are more close neighboring cases such as n_7 and n_8 in the other neighboring cluster, $Cluster_{Next}$. This phenomenon is intensified as target cases are placed closer to the boundary areas of the corresponding cluster.

In order to verify this problem, we apply the basic *C-CBR* method to the *Diabetes* dataset introduced in Section 5 and calculate the average accuracy of the target cases placed in the centered areas (90%) and the remaining boundaries (10%) separately. In this preliminary research, the results show that the average accuracy of the target cases placed in the boundary areas (0.711) is significantly lower than those of the centered areas (0.739).



(C_{Best}: The centroid of the corresponding cluster $Cluster_{Best}$
C_{Next}: The centroid of the neighbouring cluster $Cluster_{Next}$)

<Figure 2> Limitations of the Basic Clustering CBR Method

Ⅳ. The Clustering-Merging Case-Based Reasoning Method (CM-CBR)

In this section, we suggest a new hybrid Case-Based Reasoning method called the *CM-CBR* method that dynamically expands a search pool to retrieve neighbors considering the location of a target case in a corresponding cluster. Section 4.1 introduces how to determine whether or not the target cases are placed in the center areas of a cluster or in the boundary areas and Section 4.2 describes how to determine the number of clusters *k* for the *CM-CBR* method. The overall procedure of the *CM-CBR* method is explained in Section 4.3.

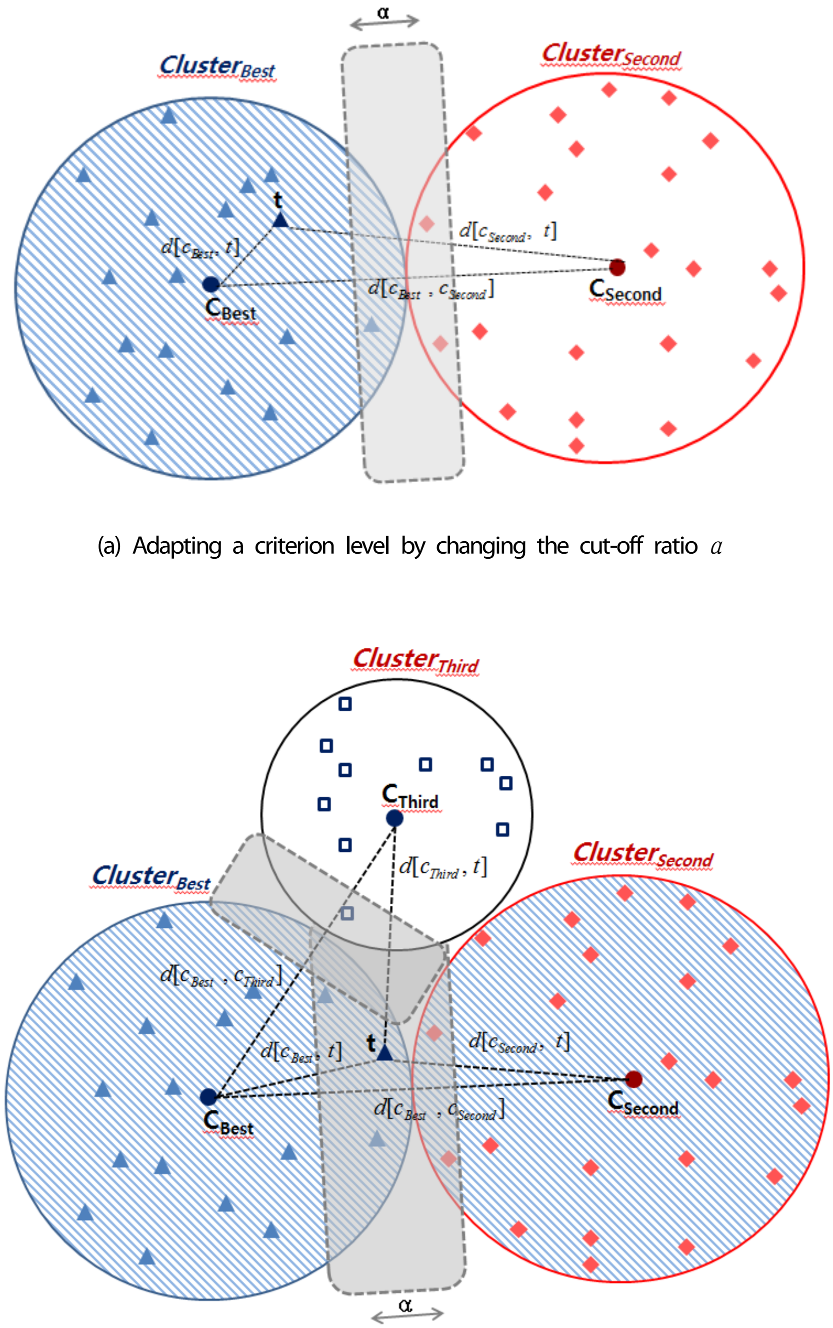
4.1. Determining the Center and Boundary Areas

The suggested *CM-CBR* method expands the searching pool to adjacent clusters for target cases placed in the boundary areas. In other words, if a target case *t* is placed in the center then the *CM-CBR* method retrieves neighbors only in a corresponding cluster like the basic *C-CBR* method. However, if *t* is placed in a *boundary area* then it searches not only the corresponding cluster $Cluster_{Best}$, but also for the other adjacent clusters, such as $Cluster_{Second}$ and $Cluster_{Third}$ to find more similar neighbors. For example, in <Figure 3 (a)>, the *CM-CBR* method uses the $Cluster_{Best}$ as the search pool to find neighbors for a center-placed target case *t*; however it expands the search pool to the $Cluster_{Next}$, from the $Cluster_{Best}$ for a boundary-placed target case *t* as depicted in <Figure 3 (b)>.

Thus, the *CM-CBR* method needs to determine whether or not the target cases are placed in the center areas of a cluster or in the boundary areas. In other to do this, we consider three different distances: first, the distance $d[C_{Best}, t]$, which calculates the distance between the target case *t* and the centroid of the corresponding cluster C_{Best}, second, the distance $d[C_{Second}, t]$, the distance between *t* and the centroid of the other close cluster C_{Second}, and lastly the distance $d[C_{Best}, C_{Second}]$, the distance between C_{Best} and C_{Second}. The Euclidian distance formula for calculating these three distances are presented as follows.

- $d[C_{Best}, t] = \sqrt{\sum(X(C_{Best}) - X(t))^2}$
- $d[C_{Second}, t] = \sqrt{\sum(X(C_{Second}) - X(t))^2}$
- $d[C_{Second}, C_{Second}] = \sqrt{\sum(X(C_{Second}) - X(C_{Second}))^2}$

Then, we set a constant a $(0 \leq a \leq 1)$ to cut-off and adjust the size of a center area and a boundary. In other words, if the difference between $d[C_{Best}, t]$ and $d[C_{Second}, t]$ is smaller than $a \times d[C_{Best}, C_{Second}]$ then the *CM-CBR* determines that the target *t* is placed in boundary areas as presented in <Figure 3 (b)>. However, if the difference is greater than a criterion, then it determines that the target *t* is placed in centered areas, such as *t* in <Figure 3 (a)>. We called this a as a cut-off ratio. In this stage, the ratio of a center to boundary area can be adjusted by changing the cut-off ratio a. As a increases the boundary areas become wider and more target cases are involved in boundary areas. However, as a decreases, more target cases are classified as placing in centered areas. In extreme cases, when α is 0, all target cases are classified as placing in a centered area, thus CM-CBR operates exactly the same as the basic C-CBR method. The optimum α, which produces good performances, for each dataset is different. Thus, it is desirable to find the optimum cut-off ratio a for each dataset. In this research, we



(a) Adapting a criterion level by changing the cut-off ratio a

(b) A searching pool is expanding to the adjacent cluster

<Figure 3> Dynamically Composing a Search Pool for Retrieving Neighbors

change the cut-off ratio (a) several times to get the adequate value which produces good performances in terms of predictive performances as well as computational costs, and set this to 0.2:

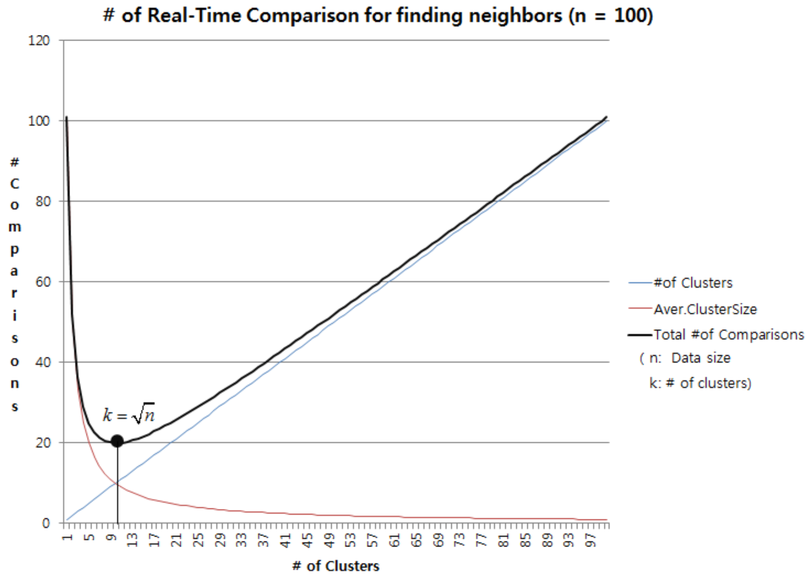
If $d[C_{Best}, t] - d[C_{Second}, t]| \le a \times d[C_{Best}, C_{Second}]|$
(a: cut-off ratio)
Then t is placed in "*boundary area*"
Else, t is placed in "*center area*"

4.2. Determining the Number of Clusters

Next, we explain how to determine the number of clusters k for the *CM-CBR* method. Usually, the basic clustering CBR (*C-CBR*) method finds the optimal k by trial and error. In other words, the *C-CBR* method is performed repeatedly in a training phase, by changing the numbers of clusters several times. After then, the k_{best}, the number of clusters that produces the best predictive performance, is selected. By contrast, the *CM-CBR* method determines the numbers of clusters focusing on minimizing case retrieval time. It is because the *CM-CBR* method can dynamically expand a searching pool for improving predictive performance, thus it is more effective to select the k, which reduces computational cost rather than improves predictive performance. In order to minimize the case retrieval time, we apply the following formula suggested in our previous work (Park, 2013):

- k= (rounding off to the nearest integer) $\sqrt{n}$
  (k: The number of clusters, n: The total number of data in a case-base)

In this work, we showed that the minimum number of computations of the *C-CBR* method are achieved when the number of clusters is calculated $\sqrt{n}$ (k= $\sqrt{n}$) as presented in <Figure 4> (Park, 2013). If the *C-CBR* method clusters a case-base into k numbers of groups, then one clustered group can contain n/k cases on average. When a new target case t comes

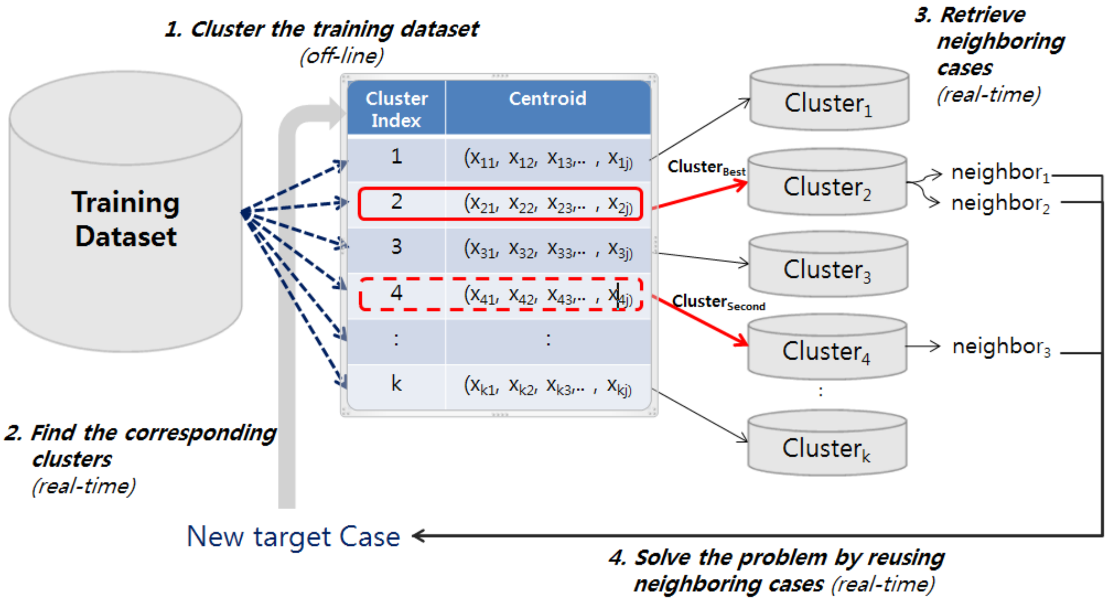


<Figure 4> The Number of Computations to Retrieve Neighbors [7]

in on-line, it finds the group that the target case is involved with among these k groups. At this stage, the *C-CBR* method calculates the distances between t and the centroids of all k clusters and selects the closest cluster ($Cluster_{Best}$) as the corresponding group, which requires k number of computations. It then searches neighboring cases of t only within the $Cluster_{Best}$, which contains n/k cases on average. Thus, n/k computations are additionally required, so the total number of computations for retrieving neighbors becomes $k + (n/k)$. We graphically present it as the number of clusters k changes in <Figure 4>. As you see, the minimum number of computations are achieved when k is $\sqrt{n}$.

4.3. The Overall Procedure of the CM-CBR

The overall procedure of the *CM-CBR* method is graphically depicted in <Figure 5> and descriptively explained in <Figure 6>. In the first step, the *CM-CBR* method performs exploratory data analysis. In the second step, it transforms data by standardization to eliminate the effects of units. Next, in the third step, the training dataset and test dataset are determined for 10 fold-cross validation. In the fourth step, the *CM-CBR* method clusters the training dataset into numbers of groups as explained in section 4.1, where n is the number of training dataset. Step 1 to 4 are performed off-line. The rest of the process starting from step 5 is performed on-line. The 5th step uses a search-pool to retrieve neighbors for a target case t with a dynamic cluster indexing technique as explained above. In the sixth step, the process retrieves neighbors from the newly composed search pool and predicts the results of a target case by using retrieved data. Steps 4-6 are repeated for 10-fold cross validation and, finally, total performance of the *CM-CBR* is calculated.



Cluster Index	Centroid
1	$(x_{11}, x_{12}, x_{13},\cdots, x_{1j})$
2	$(x_{21}, x_{22}, x_{23},\cdots, x_{2j})$
3	$(x_{31}, x_{32}, x_{33},\cdots, x_{3j})$
4	$(x_{41}, x_{42}, x_{43},\cdots, x_{4j})$
:	:
k	$(x_{k1}, x_{k2}, x_{k3},\cdots, x_{kj})$

<Figure 5> The Process of Retrieving Neighbors for CM-CBR

1. Perform exploratory data analysis (EDA); identify overall patterns and outliers.

2. Transform data for comparability:
   a. Eliminate effects of units (of measurement) by subtracting the mean and dividing by the standard deviation if the attributes are real.

3. Divide the dataset into training and test dataset for 10 fold-cross validation.

4. Cluster the training dataset into sub-groups.
   a. Cluster the training dataset by *K-Means Clustering* algorithm. The number of clusters *k* is calculated by the following formula:
   $k = \sqrt{n}$ (*n*: the number of training dataset, *k*: # of clusters, 1 <*k*<*n*)

5. Compose a search-pool to find neighbors for a target case *t*.
   a. Calculate the distance between the target case *t* and the centroid c_i of each cluster grouped in the previous step:
   $d[C_i, t] = \sqrt{\sum((X(C_i) - X(t))^2}$ (x: the value of a variable for a case
   c_i : the centroid of i_{th} cluster group, t: target case)
   b. Find the cluster having the minimum distance from a target case *t*. This cluster is set as the corresponding cluster of *t*, and referred to as the $Cluster_{Best}$. Similarly, find the second best cluster having the next smaller distance from a target case and refer to it as the $Cluster_{Second}$.
   c. Calculate the distance between the centroids of the best and the second best clusters:
   $d[C_{Best}, C_{Second}] = \sqrt{\sum(X(C_{Best}) - X(C_{Second}))^2}$ (C_{Best}: the centroid of $Cluster_{Best}$
   C_{Second}: the centroid of $Cluster_{Second}$)
   d. If $(d[C_{Best}, C_t] - d[C_{Next}, C_t]) > d[C_{Best}, C_{Next}] \times \alpha$ (*a*: a cut-off ratio, $0 \le a \le 1$)
   Then *the Best* cluster becomes the new case-base for a target case *t*.
   Else the new case-base is composed by merging the *Best* cluster and the *Next* cluster.
   e. Repeat step 5-c and 5-d until the case-base does not expand further.

6. Predict the results of a target case *t* by performing the CBR method in a search-pool composed in step 5.
   a. Retrieve the neighboring cases $x(n_j)$ in the newly composed case-base.
   b. Determine the relative weight ofj^{th} neighbor:
   $W_j = \frac{1}{J-1}[1 - \frac{d_j}{d_{TOT}}]$ ($d_{TOT} = \sum_{j=1}^{J} d_j$,, *J*: the number of neighbors
   d_j: the distance between the target case *t* and j_{th} neighboring case)
   c. Predict the result $\hat{o}(t)$ of a target case *t* as the weighted sum of output attributes of the neighboring cases:
   $\hat{o}(t) = \sum_{j=1}^{J}(W_j \times o(n_j))$ ($o(n_j)$: the value of output attribute of a neighboring case n_j.)
   d. Repeat Step 5 and Step 6 for all target cases in the test dataset.

7. Repeat Steps 4-6 for each test data set 10 times for 10-fold cross validation.

8. Evaluate the performance.

<Figure 6> The Process of the Proposed Cluster-and-Merging CBR (CM-CBR)

Ⅴ. Experiments

The common experimental settings used throughout this research are introduced in section 5.1 and the experimental results of the *CBR*, *C-CBR* and *CM-CBR* are presented in section 5.2.

5.1. Experimental Settings

In this research, three real-life medical datasets were obtained from the UCI repository. Blake and Merz are used (Blake and Merz, 1998). The datasets consist of physical records and the diagnosis results concerning patients. Physical conditions of patients are set as independent variables (IV) and the diagnosis result is set as a dependent variable (DV). The details of the datasets are given in <Table 1>.

▪ Data

- Breast Cancer: The dataset originally contained 569 examples and 32 attributes. 560 of the examples were used. The dataset consists of 2 classes where 212 cases show the presence of breast cancer and 348 cases show the absence.
- Diabetes: The dataset originally contained 768 cases and 9 attributes. 760 cases of the cases were used. The dataset consists of 2 classes where 492 cases show the presence of diabetes and 268 cases show the absence.
- Cardiotocography (CTG): The dataset consists of measurements of fetal heart rate and uterine contraction features on cardiotocograms classified by expert obstetricians and in a fetal state in patients. The dataset originally contained 2126 cases and 23 attributes; however, we only used 802 of the cases considering the proportion of each class in the dataset. The fetal state is classified into three classes where 331 cases show normal, 295 cases show suspect and 176 cases show pathologic.

▪ Clustering Method

- Clustering algorithm: The K-Means clustering algorithm is used to cluster a previous case-base into several different groups. The K-Means clustering method chooses local minimum cluster centers in the instance space via a random start iterative approximation strategy (Whitten and Frank, 2000).
- The number of clusters: The number of clusters *k* for the *C-CBR* method is determined by selecting the best performing cluster among seven different *k*. For example, 15 clusters for the *Breast Cancer* dataset, 19 clusters for the *Diabetes* dataset, and 16 clusters for the *CTG* dataset are used. However, the number of clusters for the *CM-CBR* method is set to in order to minimizes the computational cost as explained in <Figure 2>.

▪ The Performance Measurements

Two aspects of performance, predictive perform-

<Table 1> Details of the Dataset Used in the Experiment

Datasets	# Instances	# Variables	# Classes
Breast Cancer	560	31	2
Diabetes	760	9	2
CTG	802	23	3

ance and computational performance were considered to evaluate the models. To evaluate the predictive performance, we mainly used accuracy measurements. However, sensitivity and specificity were also measured for binary class datasets such as the *Breast Cancer* and *Diabetes* datasets. In order to evaluate computational performance, we used on-line computational time to predict the results of target cases. This is because one purpose of the research is reducing on-line prediction time, so called real-time, rather than the off-line computational time.

- Accuracy: The proportion of correctly classified cases.
- Sensitivity: The fraction of positive cases that are classified as positive.
- Specificity: The fraction of negative cases classified as negative.
- Prediction-time: The on-line computational time needed to predict the result of a new target case in test dataset.

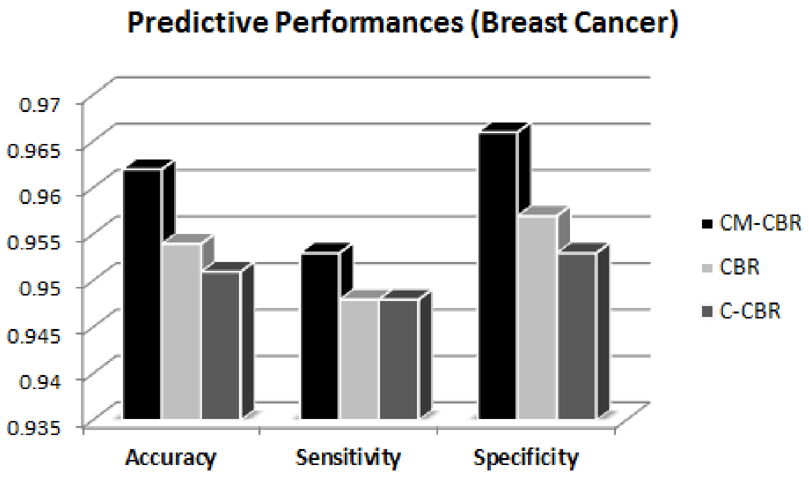
▪ Implementation

The case-based reasoning methods used throughout this research are implemented by Java and use the free data mining package Weka (Whitten and Frank, 2000). The number of neighbors used for all CBR methods is set to 3 in these experiments.

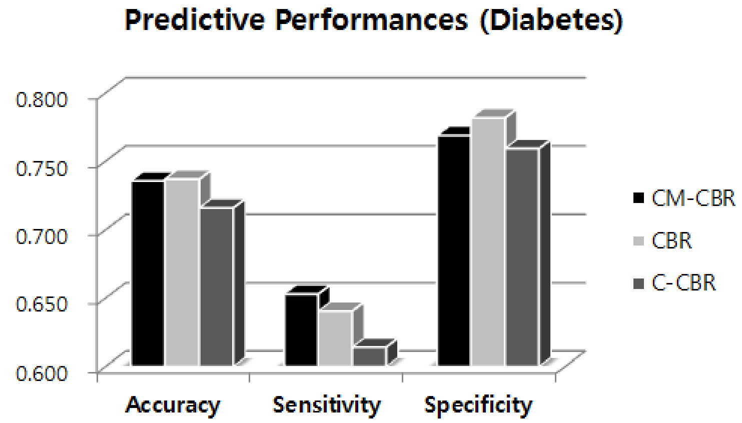
5.2. Experimental Results

This section presents the experimental results of the *CBR*, *C-CBR* and *CM-CBR* methods and compares them with each other. Aforementioned in Section 4.1, the cut-off ratio α is set to 0.2, and, in this case, 87 cases are placed in the "boundary area" for the *Breast Cancer*, 114 cases for the *Diabetes* and 120 for the *CTG* dataset are placed in the boundaries.

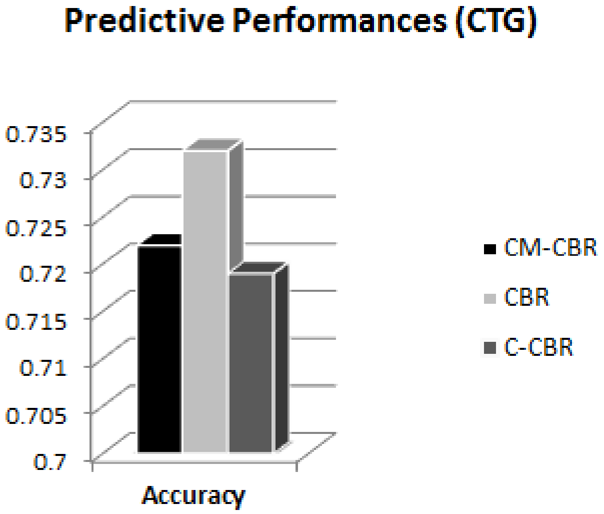
The overall predictive performances of all three methods; *CBR*, *C-CBR*, and *CM-CBR* are presented for *Breast Cancer*, *Diabetes* and *CTG* datasets in <Figure 7>, <Figure 8> and <Figure 9> respectively. In these results, the *CM-CBR* method performs the



<Figure 7> The Predictive Performances of the CM-CBR vs. CBR vs. C-CBR (Breast cancer)



<Figure 8> The Predictive Performances of the CM-CBR vs. CBR vs. C-CBR (Diabetes)



<Figure 9> The Predictive Performances of the CM-CBR vs. CBR vs. C-CBR (CTG)

best in terms of accuracy results for the *Breast Cancer* dataset and second best for the *Diabetes* and *CTG* datasets. For all three datasets, the suggested *CM-CBR* method outperforms the basic *C-CBR* method. Interestingly, the CM-CBR method produces even better predictive results than the CBR method in some cases with retrieving less proximate neighboring cases. We think there are two possibilities. First, it happens to outperform the predictive performance of the traditional CBR because of the noise in the datasets. Second, when a proximity between a target case and neighboring cases reaches to a certain level, the proximity and predictive performances are not necessarily positively correlated. The rank ordered performance of each method are also presented in <Table 2> to compare the results more efficiently.

<Table 2> Ranked Ordered Performances of Each Classifier

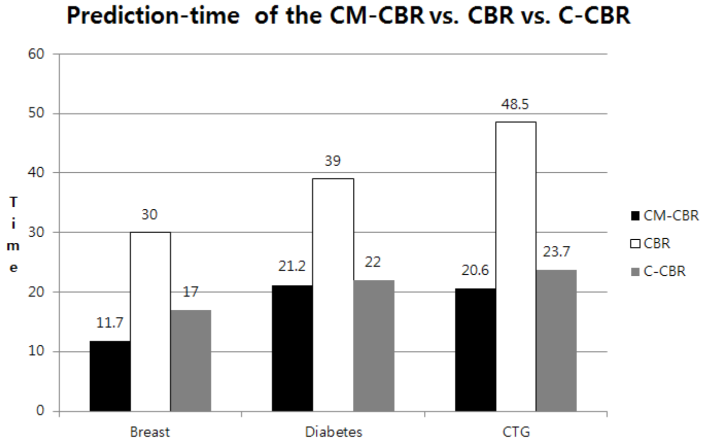
Dataset	Rank	1	2	3
Breast Cancer	Accuracy	CM-CBR 0.962	CBR 0.954	*C-CBR* 0.951
	Sensitivity	CM-CBR 0.953	CBR 0.948	*C-CBR* 0.948
	Specificity	CM-CBR 0.966	CBR 0.957	*C-CBR* 0.953
Diabetes	Accuracy	CBR 0.737	CM-CBR 0.736	*C-CBR* 0.716
	Sensitivity	CM-CBR 0.653	CBR 0.640	*C-CBR* 0.614
	Specificity	CBR 0.782	CM-CBR 0.769	*C-CBR* 0.759
CTG	Accuracy	CBR 0.732	CM-CBR 0.722	*C-CBR* 0.719

In order to verify that these results are statistically significant, we next performed one-sided paired t-tests and present the results in <Table 3>. In this experiment, the accuracy of the *CM-CBR* method statistically outperforms the basic *C-CBR* method in 4 out of 7 cases at a 90% confidence interval. Also, the *CM-CBR* method even outperforms the basic *CBR* method in 3 out of 7 cases.

Next, the real time computational costs of the *CM-CBR*, *CBR* and *C-CBR* methods are compared with each other in <Figure 10>. The unit measure of time is milliseconds. As presented in <Figure 10>, the *CM-CBR* method requires significantly less computational time to predict the results than the conventional *CBR* method. The *CM-CBR* method also requires less prediction time than the basic *C-CBR* method; however, the gap between these two methods is relatively insignificant because both methods retrieve neighbors from a pre-clustered sub groups.

<Table 3> Overview of the *t*-Test Result for Each Pair-Wised Classifier

Performance Measures	Predictive Performances					
	Accuracy		Sensitivity		Specificity	
H_0 (P-value)	*CM-CBR* $\leq$CBR	*CM-CBR* $\leq$*C-CBR*	*CM-CBR* $\leq$CBR	*CM-CBR* $\leq$*C-CBR*	*CM-CBR* $\leq$CBR	*CM-CBR* $\leq$*C-CBR*
Breast	0.0475	0.044	0.115	0.198	0.0535	0.0365
Diabetes	0.4453	0.0927	0.2670	0.0749	0.0428	0.1552
CTG	0.2815	0.4445	.	.	.	.



<Figure 10> The Prediction Time of the CM-CBR vs. CBR vs. C-CBR Methods (Milliseconds)

In conclusion, the *CM-CBR* method statistically outperforms the basic *C-CBR* method in many experimental cases in terms of predictive performance and computational cost and solves the scalability issue of the conventional *CBR* method as well.

Ⅵ. Concluding Remarks and Future Work

In this paper, we addressed the scalability issue of the conventional Case-Based Reasoning (*CBR*) method for a high-volume and rapidly increasing dataset. We also showed that, even though, the basic Clustering Case-Based Reasoning (*C-CBR*) method reduces computational costs of the *CBR* method, it often produces less accurate predictive results. Thus, practically it does not overcome the limitations of the *CBR* method.

Therefore, we suggested a new case-based reasoning method called the Clustering-Merging CBR (*CM-CBR*). The suggested *CM-CBR* method retrieves neighbors from an adaptively composed search pool considering the proximity between a target case and the centroid of a corresponding cluster. The suggested *CM-CBR* method was applied to three real-life medical datasets. The results show that it produces similar or better predictive performance than the conventional *CBR* with less computational cost. Moreover, it also outperforms the basic *C-CBR* method in terms of predictive performance.

There are some limitations in this research. First, the *CM-CBR* method is performed with the fixed cut-off ratio (i.e., 0.2) instead of applying the optimum value for each dataset. This value works well in this study; however, it may not perform well in other experimental settings. Thus, it is suggested that future researchers find the optimum cut-off ratio α for their dataset. Second, the sizes of the datasets used in the experiments are not high volume due to the difficulty of data acquisition. If the *CM-CBR* method were applied to high volume datasets in practice, the performance differences between each method would be more obvious than applying it to small size datasets. Third, we fixed some experimental settings to simplify the experiments. For example, we set the number of neighbors to 3 and calculated the distances only by the *Euclidean* method. We would like to extend this study to other experimental settings to get more general results in the future.

<References>

[1] 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.

[2] 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).

[3] Hong, T.-P., Y.-L. Liou. (2008). Case-Based Reasoning With Feature Clustering Case-Based Reasoning With Feature Clustering (2008). *7th IEEE International Conference on Cognitive Informatics,* 449-454.

[4] Khan, M. J., M. M. Awais, S. Shamail. (2008). Self-Configuration in Autonomic Systems Using Clustered CBR Approach, ICAC '08. *International Conference on Autonomic Computing,* 211-212.

[5] Kim, K.-S., I. Han. (2001). The Cluster-Indexing Method for Case-Based Reasoning Using Self-Organizing Maps and Learning Vector Quantization for Bond Rating Cases. *Expert Systems With Applications, 21*(3), 147-156.

[6] Li, Y., S. C. Shiu, S. K. Pal. (2006). Combining feature reduction and case selection in building CBR classifiers. *IEEE Transactions on Knowledge and Data Engineering, 18*(3), 415-429.

[7] Park, Y. J. (2013). A Case-Based Reasoning Method Improving Real-Time Computational Performances: Application to Diagnose for Heart Disease. *Information Systems Review, 16*(1), 37-50

[8] Park, Y. J. (2014). *Improving Real-Time Efficiency of Case-Based Reasoning for Medical Diagnosis.* Integrating Information Technology and Management for Quality of Care [ICIMTH 2014, Athens, Greece, 10-13 July 2014], 52-55.

[9] Porter, B. W., R. Bareiss, and R. C. Holte. (1990). Concept Learning and Heuristic Classification in Weak-Theory Domains. *Artificial Intelligence, 45*(1), 229-263.

[10] Qiang Y., W. Jing. (2001). Enhancing the Effectiveness of Interactive Case-Based Reasoning with Clustering and Decision Forests. *Applied Intelligence, 14*(1), 49-64.

[11] Witten, I. H., and E. Frank. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* San Francisco: Morgan Kaufmann.

◆ About the Authors ◆

Yoon-Joo Park

Yoon-Joo Park received the BS and MS degrees in computer science from Korea University and received the PhD degree in management engineering from the Korea Advanced Institute of Science and Technology. She is an assistant professor at Seoul National University of Science and Technology. Her current research interests include personalization, matching systems and text mining. Prior to starting her work at Seoul Tech, she was a manager in the IT Planning Department at Samsung Life Insurance.

Submitted: May 13, 2015; 1st Revision: July 10, 2015; Accepted: August 11, 2015