Two-Step Filtering Datamining Method Integrating Case-Based Reasoning and Rule Induction

Yoon-Joo Park[a*], Enmi Choi[a], Soo-Hyun Park[a]

[a]KOOKMIN University, 861-1 Jeongneung-Dong Seoungbuk-Gu Seoul, 136-702, Korea
Tel:82-2-583-6109, Fax: 82-2-958-3604 , E-mail :yjpark75@gmail.com
emchoi@kookmin.ac.kr
shpark@kookmin.ac.kr

Abstract

Case-based reasoning (CBR) methods are applied to various target problems on the supposition that previous cases are sufficiently similar to current target problems, and the results of previous similar cases support the same result consistently. However, these assumptions are not applicable for some target cases. There are some target cases that have no sufficiently similar cases, or if they have, the results of these previous cases are inconsistent. That is, the appropriateness of CBR is different for each target case, even though they are problems in the same domain. Thus, applying CBR to whole datasets in a domain is not reasonable. This paper presents a new hybrid datamining technique called two-step filtering CBR and Rule Induction (TSFCR), which dynamically selects either CBR or RI for each target case, taking into consideration similarities and consistencies of previous cases. We apply this method to three medical diagnosis datasets and one credit analysis dataset in order to demonstrate that TSFCR outperforms the genuine CBR and RI.

Keywords: Hybrid method; Datamining; Case-Based Reasoning; Rule Induction; Artificial Intelligence; Credit Analysis; Medical Diagnosis

1. Introduction

When humans encounter a new problem, they often try to remember similar previous experiences from the past and reuse their solutions. However, if they don't have relevant experience in the past, then they must try to solve problems using other methods, such as logical thinking or creativity. Likewise, the case-based reasoning (CBR) method solves new problems by remembering previous similar situations and reusing information and knowledge of those situations (Aamodt and Plaza, 1994). Thus, CBR is an appropriate and effective method to solve problems when previous cases are sufficiently similar to a target case and also consistently support the same results.

However, there are some target cases that do not have similar previous cases, or their results are inconsistent. For example, let us assume that we are trying to diagnose the disease of a patient. If there were two sufficiently similar previously diagnosed patients, then the classification result is simply the presence of disease. CBR is a great method for such a case. However, if there are no previously diagnosed sufficiently similar patients for comparison, then CBR is not an appropriate method, because the situation does not even satisfy the basic requirements of CBR. Likewise, let us assume that there were two sufficiently similar patients in the past and that they are exactly similar to a current patient. If one previous patient has the disease, but the other does not, then CBR cannot produce reliable results for the target patient. We can thus conclude that CBR can be an appropriate method for some target cases, but not for the others, even though they are problems in the same domain.

This article suggests a new hybrid datamining technique called the two-step filtering CBR and RI (rule induction) method (TSFCR) that dynamically selects a classifier between CBR and RI according to its appropriateness to a target case. The purpose of this article is to apply the appropriate classifiers between CBR and RI to each target case for domain problems which require explanations, such as medical diagnoses and credit analyses. The reason we decide to create a hybrid between the CBR and RI methods is that they both have explanation capabilities. We postulate that integrating these methods will provide comprehensible explanations for domain problems. In addition, their different characteristics, such as inductive inference (CBR) and deductive inference (RI), allow each method to complement the weaknesses of the other.

The main concept of TSFCR is to sift target cases that are appropriate for the CBR method using two-step

[*] Corresponding author. Tel.: +82-2-583-6109; Fax: +82-2-958-3604 ; E-mail: yjpark75@gmail.com

filtering, then applying the RI method for the target cases that fail to pass the filters. In the first step, TSFCR sifts the target cases that have sufficiently similar previous neighbors using the similarity criterion. The target cases that pass this first filter still have a chance to be evaluated by the second step; however, the other target cases that fail to pass the first step are classified by the RI method. In the second step, TSFCR evaluates the consistency of the results of previous similar cases to the selected target cases that come through the first step filter. If the previous neighbors have consistent results, then TSFCR applies CBR to them because they pass both filters. It applies RI to the target cases that fail to pass the second step filter. This is called the consistency criterion.

The suggested hybrid method, TSFCR, is applied to four real life datasets in order to verify that it outperforms CBR and RI as individual methods.

The rest of this paper is organized into four sections. Section 2 presents the research related to our study. Section 3 suggests the new hybrid datamining method called two-step filtering CBR and RI (TSFCR). Next, in Section 4, case studies in the areas of medical diagnosis and credit analysis are presented. Finally, concluding remarks and future research are discussed in Section 5.

2. Related Research

2.1. Case-Based Reasoning Methods

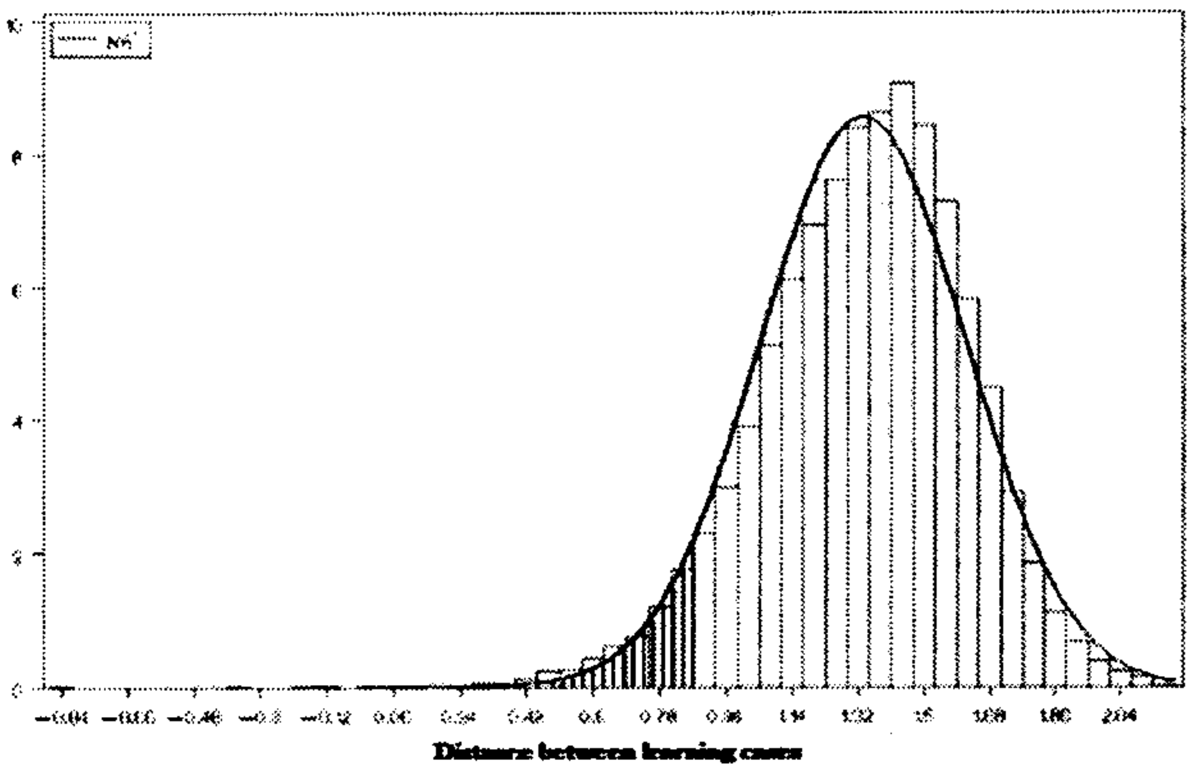
Case Based Reasoning (CBR) is an approach for solving a new problem by remembering previous similar situations and reusing information about and knowledge of those situations (Aamodt and Plaza, 1994). This concept assumes that similar problems have similar solutions, so CBR is an appropriate method for practical domains focused on real cases, and essentially performs well for target problems that have sufficiently similar previous cases. A general CBR cycle involves the following four processes identified 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.

One of the advantages of CBR is that it is relatively easy to understand in terms of how the results are produced and which cases are used for them. Another advantage of CBR is the relative ease of combining techniques with other approaches (Golding and Rosenbloom, 1996). For example, CBR has been combined with neural networks in a diagnosis system (Reategui, 1997) and also with a rule-based system for diagnosis of heart failure (Koton, 1988). However, CBR has the limitation that it is usually sensitive to noise (Cercone et al., 1999).

One of the important issues of CBR is how many neighbors to retrieve, because this number can strongly influence performance. Many previous CBR methods have tried to find out the optimal number of neighbors for a target dataset (Chun and Park, 2005). The problem with this approach is that the optimal number of neighbors is different for each target case, even in the same dataset. The recent research of Park et al. (2006) solves this problem by suggesting a new method called Statistical CBR (SCBR) that retrieves neighbors according to probabilistic similarity rather than the number of neighbors. SCBR finds out the distribution of distances between every learning case (see Figure 1), then retrieves previous neighbors that satisfy a certain cut-off probability, such as the probabilistically similar 10% at the top. Thus it can retrieve desirable neighbors for each target case. However, if there is no previous neighbor that satisfies the similarity criterion, then SCBR retrieves at least one nearest neighbor to enable performance of CBR.

Another relevant issue of CBR is how consistent the results of previous neighbors are. Conventional CBR makes decisions by comparing the integrated results of previous neighbors with the cut-off point, irrespective of the degree of adjacency between them. Park et al. (2007) suggested an interactive CBR method called Grey-Zone Case-Based Reasoning (GCBR), which makes decisions focusing additional attention on the cases that have relatively inconsistent previous results through interactive communication with users.



[Figure 1] Distribution of the distances between learning cases in the heart disease dataset

2.2. Rule Induction

Rule Induction (RI) methods learn general domain-specific knowledge from a set of training data and represent the knowledge in comprehensible form as IF-THEN rules. Most RI systems conduct heuristic searches through the hypothesis space of rules or decision trees (Cercone et al., 1999). RI is a typical deductive inference method operating by rules that people easily understand. It performs well when the cases act within the rules and there are rarely exceptions. It is also applicable for target cases that do not have specific similar previous cases, as long as they follow some rules. However, RI methods have been accused of forming only hyper-rectangular regions in the example space and not recognizing exceptions in small, low-frequency sections of the space (Cercone et al, 1999).

The decision tree is a kind of RI method. The C5.0 and classification and regression tree (CART) are well-known decision tree learners. C5.0 is an improved version of C4.5 (Quinlan, 1993) 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, 1997). CART is a binary decision tree algorithm that has exactly two branches at each internal node (Breiman et al., 1984).

2.3. Integrating CBR and RI methods

A lot of previous research has suggested hybrid methods to integrate CBR and RI. CBR is an inductive learning method, and RI is a deductive learning method; their integration can take advantage of both methods and complement their weaknesses. PROTOS uses the CBR method in a main role and rule-based reasoning (RBR) in a supporting role in the case retrieval process for CBR (Bareiss et al., 1988). CASEY also uses CBR in a main role to recall and remember problems, and uses a causal model of its domain to justify re-using previous solutions and to solve unfamiliar problems (Koton, 1989). Another previous method, INRECA, performs case-based reasoning as well as Top-Down Induction of Decision Trees (TDIDT) classification to improve both the similarity assessment and the speed of the overall system by inductive learning (Althoff et al, 1995). MCRS combines CBR and RBR to present a personnel-evaluation support system (Chi and Kiang, 1993). CABARET uses CBR in a supporting role such as aiding a cooperating inductive decision tree-based learning algorithm with training set selection, branching feature selection, deliberate bias selection and specification of inductive policy (Skalak and Rissland, 1990).

The previous research of Coenen et al. (2000) describes a method for improving response modeling by using a combined approach of RI and CBR. They apply C5.0 in the first step, ranking the classified cases by a typicality measure in the second step (Coenen et al., 2000). The research of Golding and Rosenbloom (1996) also combines both methods. Their method uses a set of rules, which are taken to be only approximately correct, to obtain a preliminary answer for a given problem. It then draws analogies from previous cases to handle exceptions to the rules (Golding and Rosenbloom, 1996). The previous research of Cercone et al. (1999) surveys major hybrid methods integrating RI and CBR systems and presents a new balanced approach to constructing a hybrid architecture, along with arguments in favor of this balance and mechanisms for achieving a proper balance (Cercone et al., 1999).

3. Two-Step Filtering CBR and RI Method

In this section, we suggest a new hybrid datamining method called two-step filtering CBR and RI (TSFCR) that integrates CBR and RI within the leading framework of CBR. The suggested method dynamically selects either CBR or RI for each target case considering the appropriateness of each method. In order to determine the necessity and validity of this research, we performed a preliminary analysis, and based on this analysis, we suggest the architecture of TSFCR. The preliminary analysis is introduced in Section 3.1, and the overall architecture and detailed algorithm of TSFCR is explained in Section 3.2.

3.1. Preliminary analysis

Before describing the architecture of TSFCR on a full scale, we outline the steps of our preliminary analysis in order to verify the necessity and validity of this research. There are two main assumptions.

The first assumption is that there are some target cases that have no previous similar cases which satisfy a certain similarity criterion, while there are others that satisfy many, although both are in the same dataset and the similarity criterion is the same. In order to verify this assumption, we count the number of neighbors that satisfy a similarity criterion for every target case in a breast cancer dataset obtained from the UCI repository (Blake and Merz, 1998) and draw a histogram of it.

1. Analyze the distribution of distances by calculating all pairwise distances between learning cases. :

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

(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)

Then we can find out the distribution of the distances D.

D~$N(\mu,\ \sigma^2)$ (D: Random variable of distances
μ : Average distance of distances
σ^2: Variance of distances)

2. Find out the previous similar neighbors X(t_i) in the learning dataset that satisfies a certain probabilistic similarity α_{opt} .:

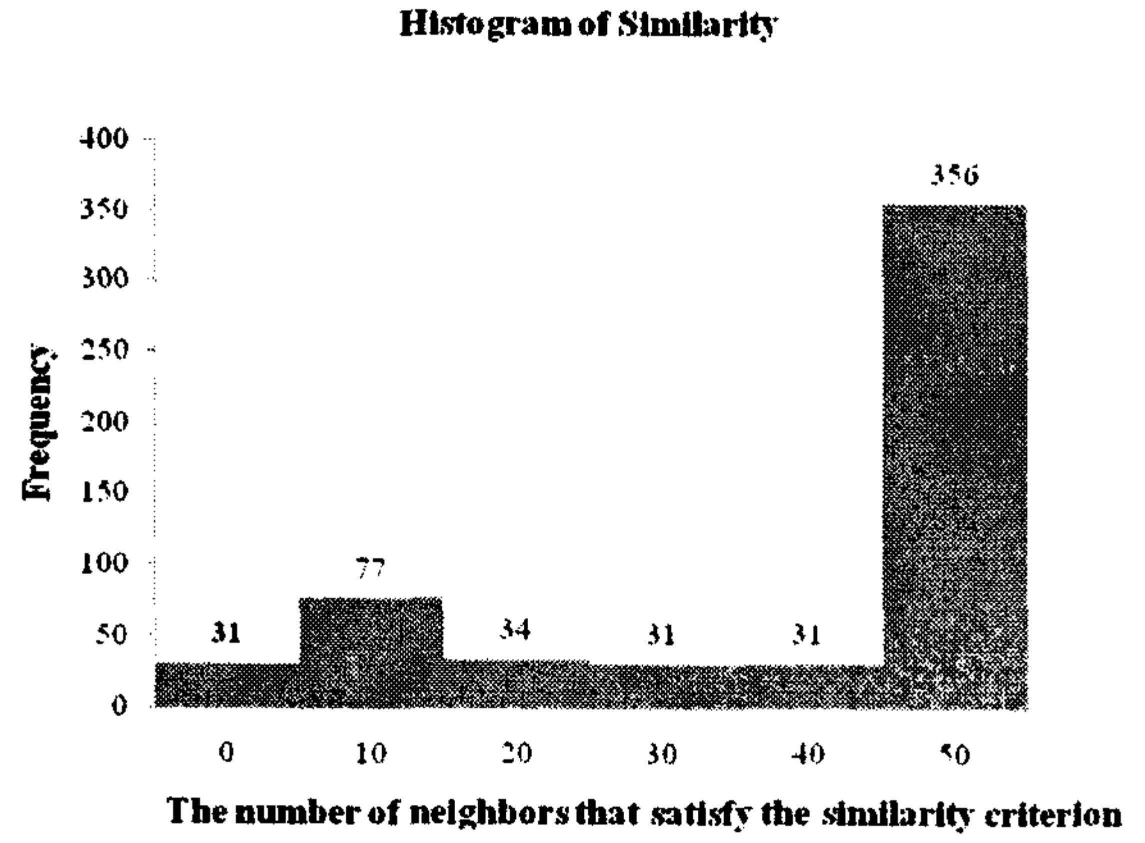
$$\alpha_i = P[Z(=\frac{d_i - S}{\sigma}) < Z_{opt}]$$

(d_i : Distance between ith neighbor and a target case X(target)
α_i : Probabilistic Similarity of ith neighbor)

[Figure 2] The algorithm to calculate probabilistic similarity

We perform 10-fold cross validation, thus there are 10 similarity criteria for each test dataset. Figure 2 provides the algorithm to calculate similarities in this experiment, and Figure 3 shows the histogram of the results. As you see in Figure 3, there is no previous similar neighbor that satisfies the similarity criterion for the 31 target cases, while there are 40 to 50 previous neighbors for the 356 target cases. The result definitely shows that the numbers of neighbors are different according to each target case, and also implies that CBR is not an appropriate method for some target cases that have no similar neighbors, like the 31 target cases in this experiment.

The second assumption is that the results of retrieved neighbors can be different, and target cases that have more consistent previous results perform better than those with less consistent results when applying CBR. The concept of consistency in this article means how often the selected neighbors produce consistent results. If every selected neighbor produces the same results, then the consistency becomes 100%; on the other hand, if there are no consistent results among the selected neighbors, so that the final result is situated at the exact classification cut-off point, then the consistency becomes 0%. We calculate consistencies using the algorithm depicted in Figure 4 and the histogram of the consistencies of retrieved neighbors for the breast cancer dataset presented in Figure 5. As you see in Figure 5, the 36 target cases have 0 to 10% consistency, while the 361 target cases have 90 to 100% consistency. We assume that the target cases that have more consistent results perform better when applying CBR. In order to verify this, we classify the total dataset into two groups by consistency and compare their accuracy. One group consists of the target cases that have more consistent results, while the others have less. Table 1 shows the average accuracies of the two groups for the diabetes, dermatology, breast cancer and credit analysis datasets, and the results indicate that the average accuracy of the more consistent groups performs better than the others for every dataset in this experiment.



[Figure 3] Histogram of the number of neighbors that satisfy the similarity criterion for the breast cancer dataset

1. Compute the sum of distances between the target case and the retrieved neighbors: $d_{TOT} = \sum_{i=1}^{J} d_i$

(d_i: The distance between ith neighbor and the target case)

J: The total number of neighboring cases)

2. Determine the relative weight of the i^{th} neighbor:

if J=1 then $w_i = 1 - \alpha_i$

else $w_i = \frac{1}{J - 1}[1 - \frac{d_i}{d_{TOT}}]$

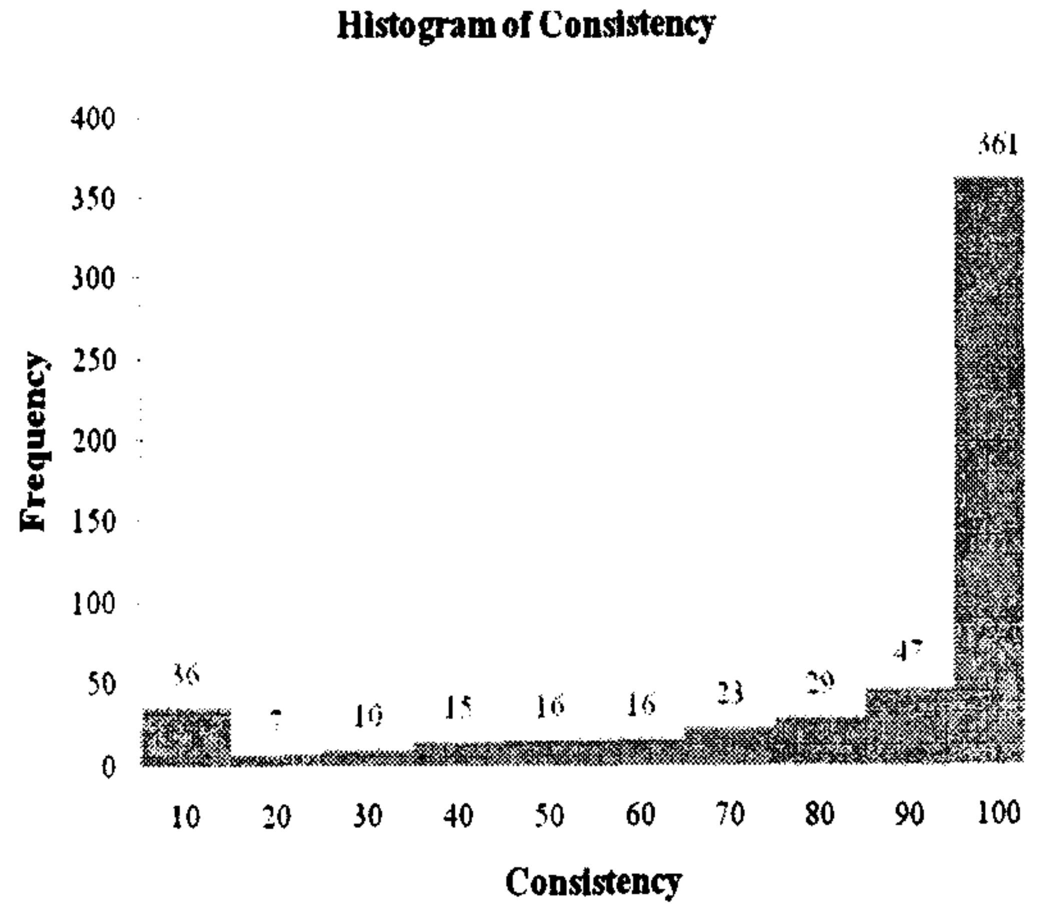
3. Sum up each weight w_i that has the same output class in $w_{class_1}, \ldots, w_{class_c}$. (n: The total number of output classes)

4. Identify the class that has the highest sum of weights w_{class}. The target case X(t) is classified as the class.

5. Calculate a consistency

$$\beta_{target.} : \beta_{target} = (w_{class} - \frac{1}{n}) * \frac{n}{n-1} * 100$$

[Figure 4] The algorithm to calculate consistency



[Figure 5] Histogram of consistency for the breast cancer dataset

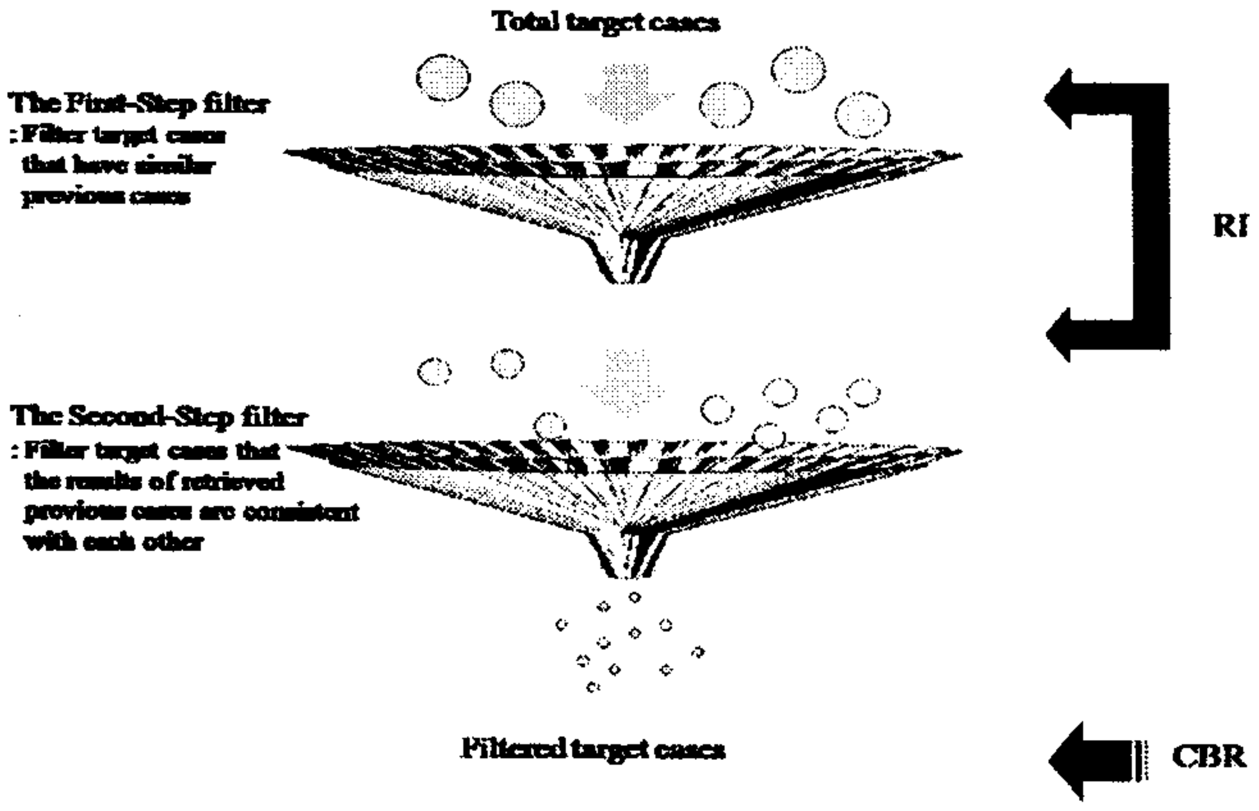
Dataset	*Diabetes*		*Dermatology*		*Breast Cancer*		*Credit Analysis*	
Consistency	*Less*	*More*	*Less*	*More*	*Less*	*More*	*Less*	*More*
# of data	427	333	53	297	121	439	466	534
Accuracy	0.64	0.89	0.74	0.99	0.83	0.98	0.57	0.84

[Table 1] Average accuracies of the less consistent and more consistent groups

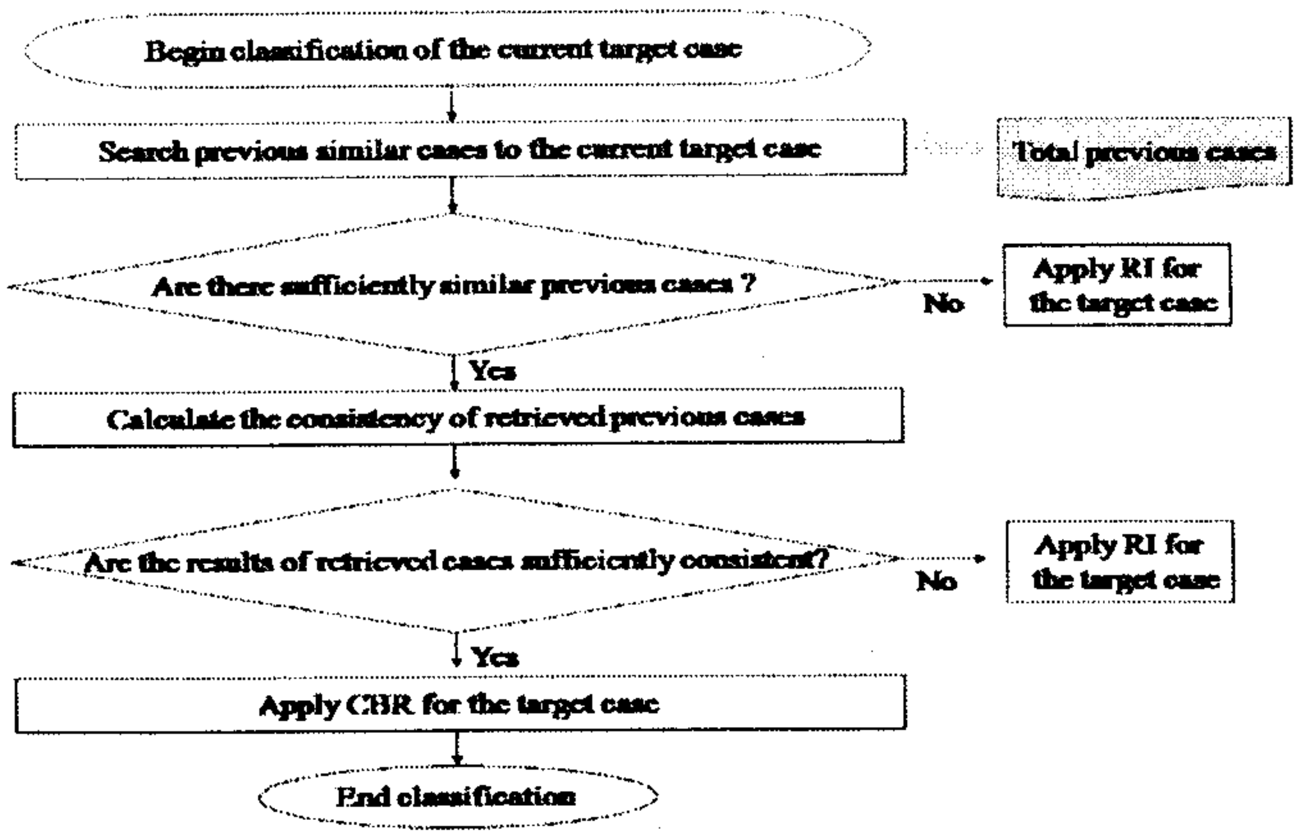
3.2. Total procedure of TSFCR

In this section we suggest the architecture and total algorithm of TSFCR. Based on the preliminary analysis, TSFCR filters out the target cases that are inadequate for CBR and applies RI to them using the two-step filter. The first-step filter operates by the similarity criterion. In this step, TSFCR selects the target cases that have sufficiently similar previous neighbors, because it is unreasonable to apply CBR to target cases that have no similar previous neighbors. The second-step filter operates by the consistency criterion. TSFCR selects the target cases with consistent results between previous neighbors for because it is desirable in applying CBR that previous neighbors strongly support the same results. Figure 6 shows a graphic depiction of the two-step filters.

The outline of TSFCR is described by the flowchart in Figure 7. First, we begin classification with the current target case X(t_{target}). Second, we search previous similar cases to X(t_{target}) in a learning dataset that satisfies a certain probabilistic similarity α_{opt}. In this stage, TSFCR calculates the similarity by the algorithm explained in Figure 2. Third, we filter out the target cases that have no neighbor that satisfies a similarity criterion, and apply RI to them. Fourth, we retrieve the similar neighbors and calculate their consistency for the target cases that pass the previous step. Fifth, we filter out the target cases whose results are inconsistent and apply RI to them. The algorithm to calculate consistency in this stage is explained in Figure 4. However, the target cases that have only one neighbor must be treated specially, since their consistency will always be 100%. Thus, in this case, TSFCR is only concerned about the similarity criterion. Finally, and sixth, we apply CBR to the target cases that pass through the two-step filters and end the TSFCR process. The total algorithm of TSFCR is described in more detail in Figure 8.


Total target cases
The First-Step filter
: Filter target cases that have similar previous cases
RI
The Second-Step filter
: Filter target cases that the results of retrieved previous cases are consistent with each other
Filtered target cases
CBR

[Figure 6] Two-step filters of TSFCR


Begin classification of the current target case
Search previous similar cases to the current target case
Total previous cases
Are there sufficiently similar previous cases ?
No
Apply RI for the target case
Yes
Calculate the consistency of retrieved previous cases
Are the results of retrieved cases sufficiently consistent?
No
Apply RI for the target case
Yes
Apply CBR for the target case
End classification

[Figure 7] The flowchart of TSFCR

Step 1. Transform data for comparability. :

a. Eliminate effects of units (of measurement) by subtracting mean and dividing by standard deviation if attributes are real type. : $V_{ij} \rightarrow ZV_{ij} \equiv Z_{ij}$

Step 2. Begin with target case $X(t_{\text{target}})$.

S tep 3. Search the neighboring cases $X(t_i)$ in the past that satisfy the certain probabilistic similarity criterion α_{opt} obtained by using the algorithm to calculate probabilistic similarity described in Figure 2.

Step 4. Filter the target cases that have any neighbor that satisfies the criterion α_{opt}, apply RI to them and stop the procedure.

S tep 5. Retrieve previous neighbors that pass the similarity criterion and calculate their consistency using the algorithm described in Figure 4.

Step 6. Filter the target cases that have consistency less than the cut-off consistency and apply RI to them. Then, apply CBR to the target cases that have consistency greater than the cut-off consistency. TSFCR determines the cut-off consistency that has the highest accuracy by changing it from 1 to 100 in the learning dataset.

[Figure 8] The total algorithm of TSFCR

4. Experiments

We investigate whether TSFCR is an effective method that performs successfully in practice by applying it to four real-life datasets and comparing the results with other genuine CBR and RI methods. The datasets used in these experiments are explained in Section 4.1, the implementation methods are described in Section 4.2, and the results of the experiments are presented in Section 4.3.

4.1. The data

We executed the case study using four real-life datasets obtained from the UCI repository (Blake and Merz, 1998).

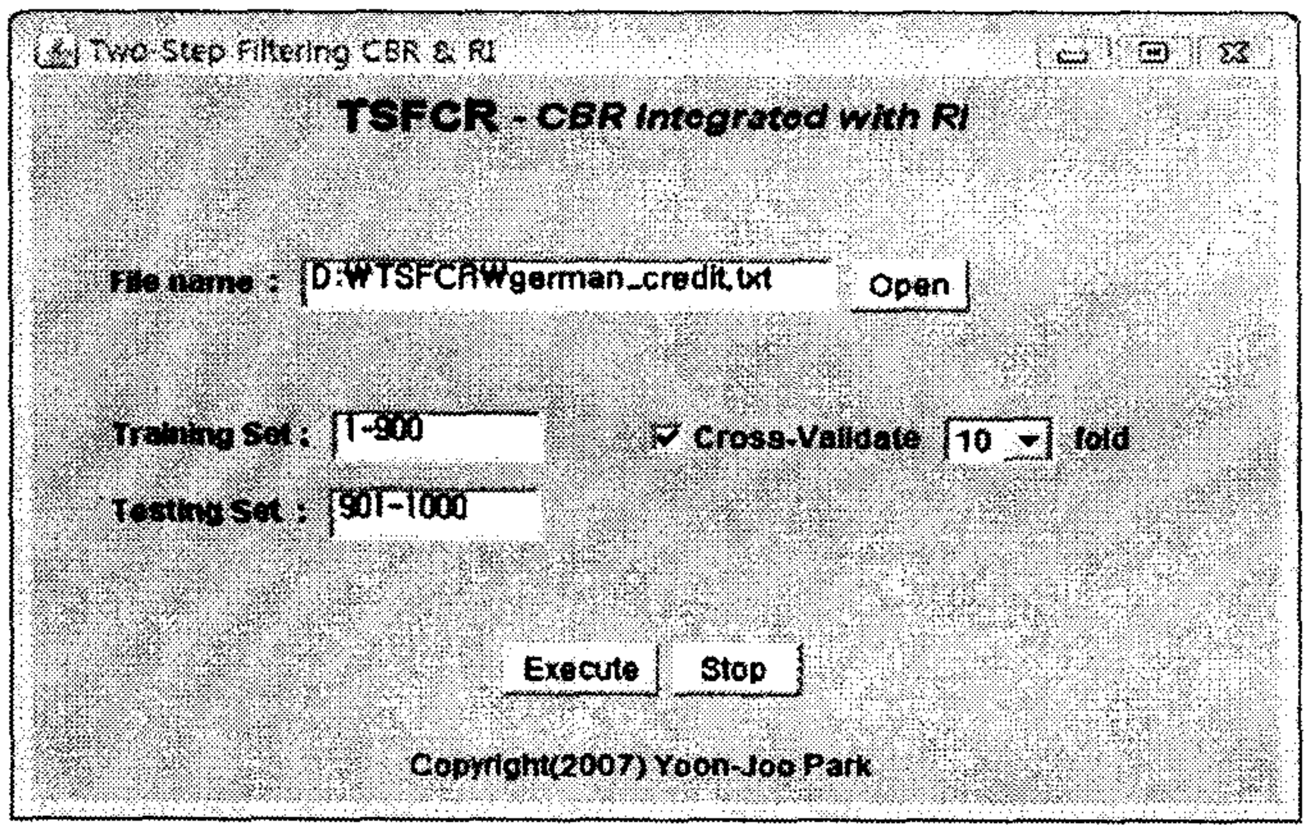
The datasets were selected to verify the effectiveness of TSFCR. They consisted of three two-class datasets and one multiclass dataset. The details of these datasets are given in Table 2.

Datasets	# Instances	# Classes	# Variables
Diabetes	760	2	9
Dermatology	350	6	35
Breast Cancer	560	2	31
Credit	1000	2	25

[Table 2] Details of the datasets used in the experiment

4.2. Implementation

We implemented CBR and TSFCR using JAVA and the commercial application Clementine8.5 for RI. In this experiment we also used C5.0 as a RI method, since it is the advanced version of C4.5 and it performs better than CART when integrated with CBR based on empirical results. Figure 9 shows the initial interface of TSFCR, and Figure 10 shows the result summary. The result summary provides detailed information about classification, such as which method is used for a target case, reasons to use this method, classification results and whether it is correct or wrong. For example, target case 903 in Figure 10 has 9 previous neighbors that satisfy the similarity criterion 19%; thus it passes the first-step filter. However, the consistency of those neighbors is 32.3%, which is less than the cut-off consistency 43%. Thus. it fails to pass the second-step filter. So, C5.0 is used for classifying target case 903 instead of CBR. The result of case 903 is "good credit" and the real result is also the "good credit", so the classification of TSFCR is correct. If the "Show rules" button is pushed in Figure 10, the new interface pops up like in Figure 11, and it shows the specific rules that are used for the classification.


Two-Step Filtering CBR & RI
TSFCR - CBR Integrated with RI
Open
Execute
Stop
Copyright(2007) Yoon-Joo Park

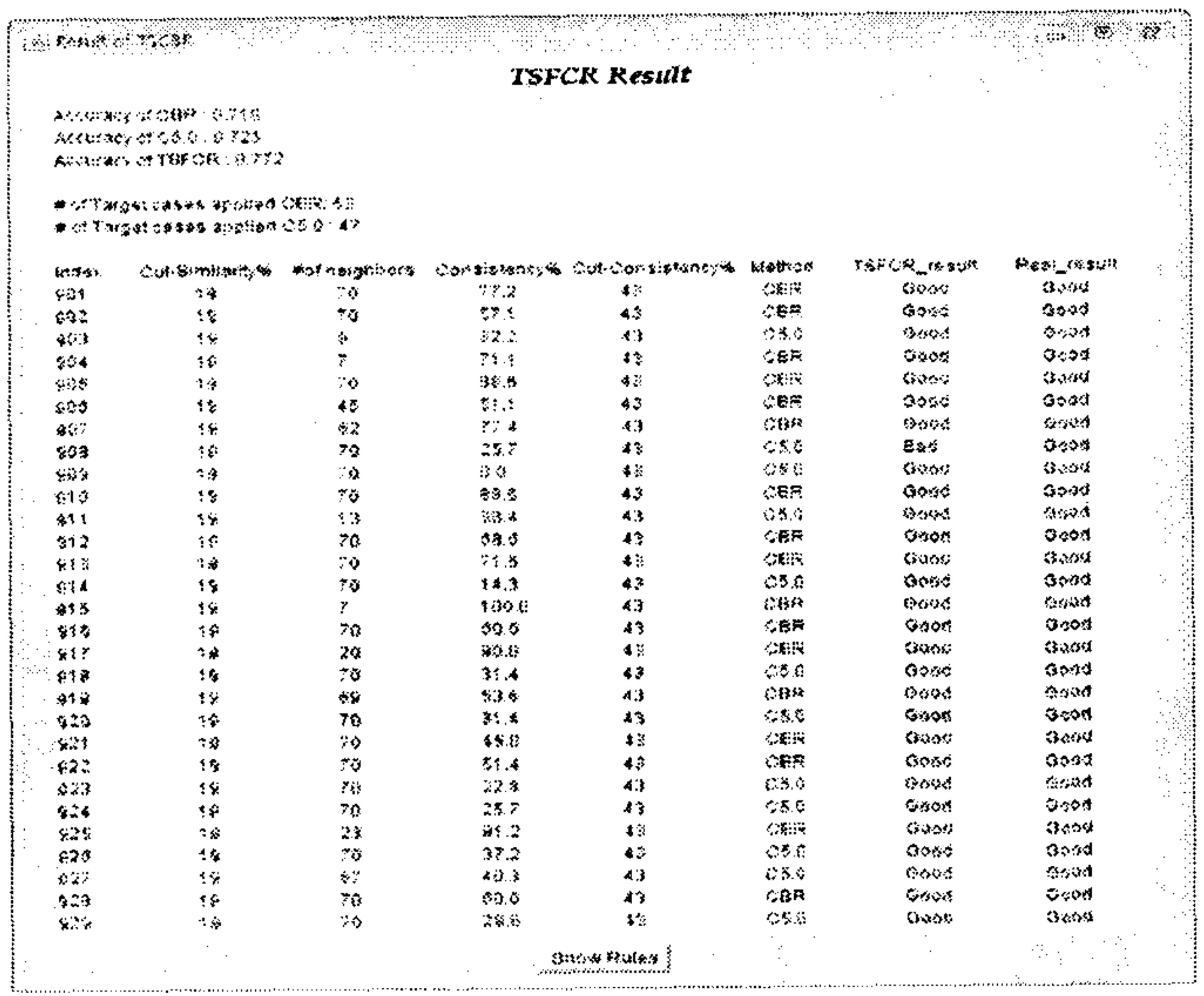
File name : D:₩TSFCR₩german_credit.txt

Training Set : 1-900

☑ Cross-Validate 10 fold

Testing Set : 901-1000

[Figure 9] Initial interface of TSFCR

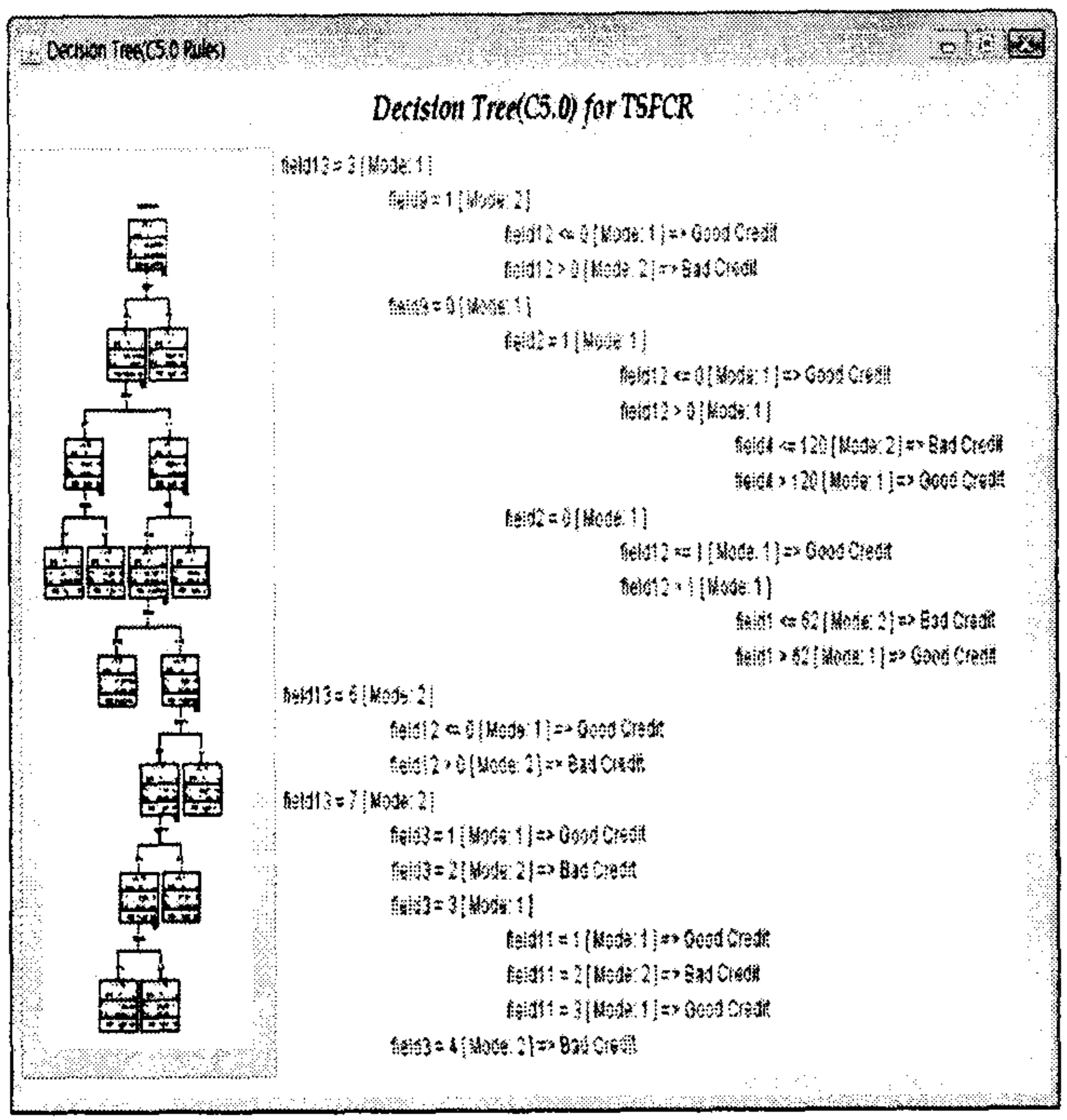


Accuracy of CBR : 0.715
Accuracy of C5.0 : 0.723
Accuracy of TSFCR : 0.772

# of Target cases applied CBR: 53
# of Target cases applied C5.0 : 47

Index	Cut-Similarity%	#of neighbors	Consistency%	Cut-Consistency%	Method	TSFCR_result	Real_result
901	19	70	77.2	43	CBR	Good	Good
902	19	70	57.1	43	CBR	Good	Good
903	19	9	32.3	43	C5.0	Good	Good
904	19	7	71.4	43	CBR	Good	Good
905	19	70	98.6	43	CBR	Good	Good
906	19	45	51.1	43	CBR	Good	Good
907	19	62	77.4	43	CBR	Good	Good
908	19	70	25.7	43	C5.0	Bad	Good
909	19	70	0.0	43	C5.0	Good	Good
910	19	70	88.6	43	CBR	Good	Good
911	19	13	38.4	43	C5.0	Good	Good
912	19	70	68.6	43	CBR	Good	Good
913	19	70	71.5	43	CBR	Good	Good
914	19	70	14.3	43	C5.0	Good	Good
915	19	7	100.0	43	CBR	Good	Good
916	19	70	60.0	43	CBR	Good	Good
917	19	20	90.0	43	CBR	Good	Good
918	19	70	31.4	43	C5.0	Good	Good
919	19	69	53.6	43	CBR	Good	Good
920	19	70	31.4	43	C5.0	Good	Good
921	19	70	45.0	43	CBR	Good	Good
922	19	70	51.4	43	CBR	Good	Good
923	19	70	22.9	43	C5.0	Good	Good
924	19	70	25.7	43	C5.0	Good	Good
925	19	23	91.2	43	CBR	Good	Good
926	19	70	37.2	43	C5.0	Good	Good
927	19	67	40.3	43	C5.0	Good	Good
928	19	70	60.0	43	CBR	Good	Good
929	19	70	28.6	43	C5.0	Good	Good

[Figure 10] Result summary for TSFCR



```
field13 = 3 [ Mode: 1 ]
        field9 = 1 [ Mode: 2 ]
                field12 <= 0 [ Mode: 1 ] => Good Credit
                field12 > 0 [ Mode: 2 ] => Bad Credit
        field9 = 0 [ Mode: 1 ]
                field2 = 1 [ Mode: 1 ]
                        field12 <= 0 [ Mode: 1 ] => Good Credit
                        field12 > 0 [ Mode: 1 ]
                                field4 <= 120 [ Mode: 2 ] => Bad Credit
                                field4 > 120 [ Mode: 1 ] => Good Credit
                field2 = 0 [ Mode: 1 ]
                        field12 <= 1 [ Mode: 1 ] => Good Credit
                        field12 > 1 [ Mode: 1 ]
                                field1 <= 62 [ Mode: 2 ] => Bad Credit
                                field1 > 62 [ Mode: 1 ] => Good Credit
field13 = 6 [ Mode: 2 ]
        field12 <= 0 [ Mode: 1 ] => Good Credit
        field12 > 0 [ Mode: 2 ] => Bad Credit
field13 = 7 [ Mode: 2 ]
        field3 = 1 [ Mode: 1 ] => Good Credit
        field3 = 2 [ Mode: 2 ] => Bad Credit
        field3 = 3 [ Mode: 1 ]
                field11 = 1 [ Mode: 1 ] => Good Credit
                field11 = 2 [ Mode: 2 ] => Bad Credit
                field11 = 3 [ Mode: 1 ] => Good Credit
        field3 = 4 [ Mode: 2 ] => Bad Credit
```

[Figure 11] Decision tree used for TSFCR

4.3. Experimental results

In this section, we show the results of the experiments. We compare the accuracy of C5.0, CBR and TSFCR. The results of CBR can be changed as the number of neighbors changes, so we retrieve previous neighbors by probabilistic similarity using SCBR, as suggested by Park et al. (2006).

The overall accuracies of TSFCR, C5.0 and CBR for the diabetes, dermatology, breast cancer and credit analysis datasets are presented in Tables 3, 4, 5 and 6, respectively. We performed 10-fold cross-validation; thus, there are 10 different cut-off similarities and cut-off consistencies for each fold which TSFCR uses for classifications. For example, in the experiment using the diabetes dataset at the first group displayed in Table 3, TSFCR retrieves the previous neighbors in the top 21% in terms of similarity and calculates their consistency. The cut-off consistency level which determines whether to apply CBR or RI is 56%. TSFCR uses these similarity and consistency criteria to decide whether to apply CBR or RI with an accuracy of 0.8421. Table 7 shows the rank ordered accuracy of each classifier. The average accuracy of TSFCR is the highest among the three classifiers for every dataset in these experiments. We also performed a t-test to verify that the results were statistically significant (see Table 8). The average accuracy of TSFCR is significantly better than that of C5.0 at the 95% confidence interval for the diabetes, dermatology, breast cancer and credit datasets. Likewise, it is significantly better than CBR at 95% for the diabetes and credit datasets and different from CBR at 90% for the dermatology and breast cancer datasets.

Fold #	*Cut_Similarity*	*Cut_Consistency*	*Accuracy*		
			TSFCR	*CBR*	*C5.0*
1	21%	56%	0.8421	0.7763	0.7368
2	11%	56%	0.8684	0.7763	0.7500
3	12%	56%	0.7763	0.7632	0.7105
4	21%	56%	0.7500	0.7237	0.6711
5	24%	56%	0.7368	0.7500	0.7105
6	22%	56%	0.8026	0.7237	0.7368
7	21%	56%	0.8421	0.8289	0.8421
8	8%	56%	0.8289	0.6974	0.8158
9	15%	56%	0.8553	0.7500	0.7500
10	12%	56%	0.7763	0.6974	0.8026
Average			0.8079	0.7487	0.7526
St.dev.			0.0461	0.0404	0.0529

[Table 3] Accuracy (Diabetes)

Fold #	*Cut_Similarity*	*Cut_Consistency*	*Accuracy*		
			TSFCR	*CBR*	*C5.0*
1	11%	52%	0.9714	1.0000	0.8857
2	17%	50%	0.9429	0.8857	0.8571
3	11%	52%	1.0000	0.9714	0.9714
4	22%	52%	0.9429	0.9429	0.9143
5	7%	52%	1.0000	0.9714	0.8857
6	5%	52%	0.9429	0.9714	0.8857
7	18%	52%	1.0000	0.9143	0.9714
8	8%	52%	1.0000	1.0000	1.0000
9	11%	52%	1.0000	0.9714	1.0000
10	11%	52%	0.9714	0.8857	1.0000
Average			0.9771	0.9514	0.9371
St.dev.			0.0263	0.0427	0.0568

[Table 4] Accuracy (Dermatology)

Fold #	*Cut_Similarity*	*Cut_Consistency*	*Accuracy*		
			TSFCR	*CBR*	*C5.0*
1	24%	56%	0.9643	0.9464	0.9286
2	17%	56%	0.9286	0.9286	0.8929
3	17%	76%	0.9464	0.9286	0.9286
4	17%	76%	0.9643	0.9643	0.9286
5	21%	76%	0.9821	0.9643	0.9821
6	17%	56%	0.9286	0.9286	0.9286
7	17%	76%	1.0000	0.9821	0.9286
8	24%	76%	1.0000	0.9643	0.9643
9	23%	76%	0.9821	0.9286	0.9464
10	17%	76%	0.9643	0.9464	0.9464
Average			0.9661	0.9482	0.9375
St.dev.			0.0259	0.0197	0.0242

[Table 5] Accuracy (Breast cancer)

Fold #	*Cut_Similarity%*	*Cut_Consistency%*	*Accuracy*		
			TSFCR	*CBR*	*C5.0*
1	15%	38%	0.7900	0.7300	0.7300
2	21%	48%	0.6900	0.7000	0.6800
3	24%	51%	0.8000	0.7300	0.7000
4	21%	43%	0.7400	0.7100	0.6900
5	22%	43%	0.8300	0.6900	0.7500
6	15%	48%	0.7000	0.6600	0.7300
7	18%	43%	0.7900	0.7100	0.7200
8	25%	55%	0.7600	0.7500	0.7200
9	22%	38%	0.8200	0.7300	0.7700
10	19%	43%	0.8000	0.7700	0.7600
Average			0.7720	0.7180	0.7250
St.dev.			0.0483	0.0312	0.0295

[Table 6] Accuracy (Credit analysis)

	Rank		
Dataset	1	2	3
Diabetes	**TSFCR**	C5.0	CBR
(Aver. Accuracy)	(0.8079)	(0.7526)	(0.7487)
Dermatology	**TSFCR**	CBR	C5.0
(Aver. Accuracy)	(0.9771)	(0.9514)	(0.9371)
Breast Cancer	**TSFCR**	CBR	C5.0
(Aver. Accuracy)	(0.9661)	(0.9482)	(0.9375)
Credit	**TSFCR**	C5.0	CBR
(Aver. Accuracy)	(0.772)	(0.725)	(0.718)

[Table 7] Ranked ordered accuracy of each classifier

	P values			
Hypothesis	*Diabetes*	*Dermatology*	*Breast Cancer*	*Credit Analysis*
TSFCR-CBR>0	**0.0034**	**0.0628**	**0.0502**	**0.0048**
TSFCR-C5.0>0	**0.0114**	**0.0322**	**0.0100**	**0.0095**

[Table 8] Overview of the t-test results for each pairwise classifier

5. Conclusion

We proposed a new hybrid datamining method that is able to apply dynamically an appropriate classifier between CBR and RI for each target case in this article. We ascertained that the appropriateness of the CBR method for each target case can be different, even though problems within a domain are similar according to preliminary analysis. Thus, in order to select the best method for each target case, we suggest a two-step filtering CBR and RI method (TSFCR). TSFCR classifies the target cases appropriate to apply either CBR or RI using a two-step filter, similarity filter and consistency filter. We apply this method to four real-life datasets that need explanation in the areas of medical diagnosis and credit analysis. The experimental results show that the average accuracy of TSFCR is significantly better than that of CBR and C5.0 in many cases. The limitation of this research is that it is unable to guarantee the appropriateness of RI. Thus, our future work to complement the present study will include evaluation of the adequacy of RI and applying more accurate and reliable classification for every single target case.

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.

Althoff, K., S. Wess, R. Traphoner (1995) INRECA: A seamless integration of induction and case-based reasoning for decision support tasks, *Proceedings of the eighth workshop German SIGN on machine learning.*

Bareiss, E. Ray, B. W. Porter, C. C. Wier (1988) Protos: An exemplar-based learning apprentice , *International journal of man-machine studies*, 29(5), 549-561.

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., J. H. Friedman, R. A. Olshen, C. J. Stone (1984) Classification and regression trees, *Wadsworth and Brooks*, Monterey, CA.

Cercone, N., A. An, C. Chan (1999) Rule-induction and case-based reasoning: Hybrid architectures appear advantageous, *IEEE transactions on knowledge and data,* 11(1), January/February.

Chi, R. T. H., M. Y. Kiang (1993) Reasoning by coordination: An integration of case-based and rule-based reasoning systems, *Knowledge-based systems*, 6(2), 103-113.

Coenen , F., G. Swinnen, K. Vanhoof, G. Wets (2000) The improvement of response modeling: Combining rule-induction and case-based reasoning, *Expert systems with applications*, 18, 307–313.

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

Golding, A. R., P. S. Rosenbloom (1996) Improving rule-based systems through case-based reasoning, *Artificial intelligence in medicine*, 87(1/2), 215-254.

Koton, P. (1988) Reasoning about evidence in causal explanations, *Case-based reasoning*, 1988, 260–270.

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.

Park, Y.-J., B.-C. Kim (2007) An interactive case-based reasoning method considering proximity from the cut-off point, *Expert systems with applications*, doi:10.1016/j.eswa.2006.08.003.

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

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

Reategui, E. B., J. A. Campbell, B. F. Leao (1997) Combining a neural network with case-based reasoning in a diagnostic system, *Artificial intelligence in medicine*, 9(1), 5-27.

Skalak, D. B., E. L. Rissland (1990) Inductive learning in a mixed paradigm setting, *Proceedings AAAI-90,* 840-847.