A Fuzzy-Rough Approach for Case Base Maintenance

Guoqing Cao, Simon Shiu and Xizhao Wang

Department of Computing, Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{csgqcao, csckshiu, csxzwang}@comp.polyu.edu.hk

Abstract. This paper proposes a fuzzy-rough method of maintaining Case-Based Reasoning (CBR) systems. The methodology is mainly based on the idea that a large case library can be transformed to a small case library together with a group of adaptation rules, which take the form of fuzzy rules generated by the rough set technique. In paper [1], we have proposed a methodology for case base maintenance which used a fuzzy decision tree induction to discover the adaptation rules; in this paper, we focus on using a heuristic algorithm, i.e., a fuzzy-rough algorithm [2] in the process of simplifying fuzzy rules. This heuristic, regarded as a new fuzzy learning algorithm, has many significant advantages, such as rapid speed of training and matching, generating a family of fuzzy rules which is approximately simplest. By applying such a fuzzy-rough learning algorithm to the adaptation mining phase, the complexity of case base maintenance is reduced, and the adaptation knowledge is more compact and effective. The effectiveness of the method is demonstrated experimentally using two sets of testing data, and we also compare the maintenance results of using fuzzy ID3, in [1], and the fuzzy-rough approach, as in this paper.

1 Introduction

At present, large-scale CBR systems are becoming more popular, with case-library sizes ranging from thousands [3][4] to millions of cases [5]. Large case library sizes raise problems of case retrieval efficiency, and many CBR researchers pay more attention to the problem of Case Base Maintenance (CBM). According to Leake and Wilson [6] "case base maintenance is the process of refining a CBR system's case base to improve the system's performance". That is, "case base maintenance implements policies for revising the organization or contents (representation, domain content, accounting information, or implementation) of the case base, in order to facilitate future reasoning for a particular set of performance objectives".

How should we maintain large case-based reasoning systems? In the past, researchers have done much work in this area. Smyth and Keane [7] suggested a competence-preserving deletion approach. They put forward the concept of competence (or coverage), being the range of target problems that a given system can solve and also a fundamental evaluation criterion of CBR system performance. Smyth and McKenna [8] also presented a new model of case competence, and demonstrated a way in which the proposed model of competence can be used to assist case authors.

Anand et al. [9] proposed to use data mining techniques for mining adaptation knowledge, and maintaining CBR systems.

Recently, Richter proposed the notion of knowledge containers [10][11], and it quickly became the standard paradigm for representation of the structural elements in CBR systems. Simon et al. established a methodology that could be used to transfer case knowledge to adaptation knowledge [1]. The methodology integrated identifying salient features, distinguishing different concepts, learning adaptation knowledge, computing case competence, and selecting representative cases together into a framework of CBM. Fuzzy set theory, as proposed by L.A. Zadeh [12], and rough set theory, allow the utilization of uncertain knowledge by means of fuzzy linguistic terms and their membership functions, which reflects human's understanding of the problem [13]. The rough set theory proposed by Z. Pawlak [14] enables us to find relationships between data without any additional information such as prior probability, only requiring knowledge representation as a set of if-then rules [13]. In this paper, we propose a new method of adaptation knowledge discovery, integrating rough set theory and fuzzy set theory to transfer the case knowledge to adaptation knowledge. This fuzzy-rough approach has many significant advantages, such as rapid speed of training and matching, generating a family of fuzzy rules which is approximately simplest. By applying such a fuzzy-rough learning algorithm to the phase of mining adaptation rules, the cost and complexity of case base maintenance is reduced, and the more important virtue is that the adaptation knowledge is more compact, effective and easily used.

2 Methodology for CBM using Fuzzy-Rough Approach

In this paper, we use the framework of case base maintenance in [1] to carry out our CBM process. The details of maintaining a case-base from scratch, as proposed in [1], consists of four phases: firstly, an approach to learning feature weight automatically is used to evaluate the importance of different features in a given case base; secondly, clustering of cases will be carried out to identify different concepts in the case base using the acquired feature knowledge; thirdly, adaptation rules will be mined for each concept using fuzzy decision trees, but in this paper, we apply a fuzzy-rough approach to mine adaptation rules for each concept; finally, a selection strategy based on the concepts of e -coverage and e -reachability is used to select representative cases.

In the following sub-section, we briefly introduce phases 1, 2 and 4 of the methodology proposed in [1], and introduce our approach to step 3 in detail.

2.1 Phase One - Learning Feature Weights

In this section, a feature evaluation function is defined. The smaller the evaluation value, the better the corresponding features. Thus we would like to find the weights such that the evaluation function attains its minimum. The task of minimization of the evaluation function with respect to weights is performed using a gradient descent technique. We formulate this optimization problem as follows:

For a given collection of feature weights $w_j \left(w_j \in [0,1],\ j = 1, \cdots n \right)$ and a pair of cases e_p and e_q, equation (1) defines a weighted distance measure $d_{pq}^{(w)}$ and equation (2) defines a similarity measure $SM_{pq}^{(w)}$.

$$d_{pq}^{(w)} = d^{(w)}(e_p, e_q) = \left(\sum_{j=1}^{n} w_j^2 (x_{pj} - x_{qj})^2 \right)^{1/2} = \left(\sum_{j=1}^{n} w_j^2 \boldsymbol{c}_j^2 \right)^{1/2} \tag{1}$$

where $\boldsymbol{c}_j^2 = (x_{pj} - x_{qj})^2$. When all the weights are equal to 1, the distance metric defined above degenerates to the Euclidean measure, denoted by $d_{pq}^{(1)}$, in short, denoted by d_{pq}.

$$SM_{pq}^{(w)} = \frac{1}{1 + \boldsymbol{a} \cdot d_{pq}^{(w)}} \tag{2}$$

where $\boldsymbol{a}$ is a positive parameter. When all the weights take value 1, the similarity measure is denoted by $SM_{pq}^{(1)}$.

A feature evaluation index E is defined as

$$E(w) = \frac{2 * [\sum_{p} \sum_{q(q<p)} \left(SM_{pq}^{(w)} (1 - SM_{pq}^{(1)}) + SM_{pq}^{(1)} (1 - SM_{pq}^{(w)} \right)]}{N * (N-1)} \tag{3}$$

where N is the number of cases in the case base.

To minimize equation (3), we use a gradient descent technique. The change in w_j (i.e. Δw_j) is computed as

$$\Delta w_j = -\boldsymbol{h} \frac{\partial E}{\partial w_j}, \tag{4}$$

for $j = 1, \cdots, n$, where $\boldsymbol{h}$ is the learning rate.

The training algorithm is described as follows:

Step 1. Select the parameter α and the learning rate η.

Step 2. Initialize w_j with random values in [0, 1].

Step 3. Compute Δw_j for each j using equation (4).

Step 4. Update w_j with $w_j + \Delta w_j$ for each j.

Step 5. Repeat step 3 and step 4 until convergence, i.e., until the value of E becomes less than or equal to a given threshold or until the number of iterations exceeds a certain predefined number.

2.2 Phase Two - Partitioning the Case Library into Several Clusters

This section attempts to partition the case library into several clusters by using the weighted distance metric with the weights learned in section 2.1. Since the considered features are considered to be real-valued, many methods, such as K-Means clustering [15] and Kohonen's self-organizing network [16], can be used to partition the case library. However in this paper, in order to compare the fuzzy decision tree and fuzzy-rough approaches in mining adaptation rules, we take the similarity matrix clustering method in [1].

2.3 Phase Three - Mining Adaptation Rules by Fuzzy-Rough Approach

For each cluster $L = \{e_1, e_2, \cdots, e_m\}$, we denote its cases in the form of $e_i = (x_{i1}, x_{i2}, \cdots, x_{in}, c_i)$, where x_{ij} corresponds to the value of feature $F_j (1 \le j \le n)$ and c_i corresponds to the action $(i = 1, \cdots, m)$. Arbitrarily taking a case $e_k (1 \le k \le m)$ in the cluster L, a set of vectors, namely $\left\{f_i \mid f_i \in R^{n+1}, i = 1,2,\cdots,m\right\}$, can be computed in the following way:

$$f_i = e_i - e_k = (x_{i1} - x_{k1}, x_{i2} - x_{k2}, \cdots, x_{in} - x_{kn}, c_i - c_k) =$$
$$\{y_{i1}, y_{i2}, \cdots, y_{in}, u_i\}$$

We attempt to find several adaptation rules with respect to the case $e_k (1 \le k \le m)$ from the set of vectors $\left\{f_i \mid f_i \in R^{n+1}, i = 1,2,\cdots,m\right\}$ by fuzzy rules.

Consider a problem of learning from examples in which there are n+1 numerical attributes, $\left\{Attr^{(1)}, Attr^{(2)}, \cdots, Attr^{(n)}, Attr^{(n+1)}\right\}$ ($Attr^{(n+1)}$) is the classification attribute). Then $\{f_i \mid i = 1,2,\cdots,m\}$ can be regarded as m examples described by the n+1 attributes. We first fuzzify these n+1 numerical attributes into linguistic terms.

The number of linguistic terms for each attribute is assumed to be five (which can be enlarged or reduced if it is needed in a real problem). These five linguistic terms are Negative Big, Negative Small, Zero, Positive Small, and Positive Big, in short, NB,NS, ZE, PS and PB respectively. Their membership functions are supposed to have triangular form and are shown in Figure 1. For each attribute (the k-th attribute $Attr^{(k)}$, $1 \le k \le n+1$) with the attribute-values $Range\ (Attr^{(k)}) = \{y_{1k}, y_{2k}, \cdots, y_{mk}\}$, the two parameters in Figure 1, a and b, are defined by

$$a = \Sigma_{y \in N}\, y / Card(N) \quad \text{and} \quad b = \Sigma_{y \in P}\, y / Card(P) \qquad (5)$$

in which $N = \left\{ y \mid y \in Rang\ (Attr^{(k)}), y < 0 \right\}$, $P = Range\ (Attr^{(k)}) - N$ and Card(E) denotes the cardinality of a crisp set E.

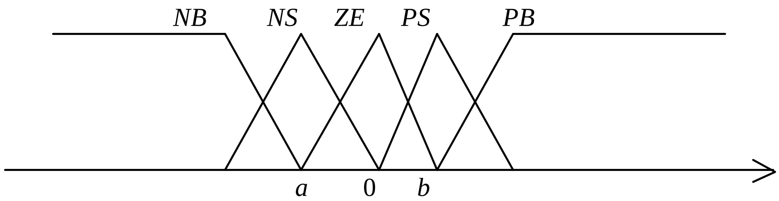

Figure 1. Five membership functions

After the process of fuzzification, we transform the crisp cases in the case library to fuzzy cases successfully. Each fuzzy case is considered to be a fuzzy set defined on the non-fuzzy label space consisting of all values of attributes, where the non-fuzzy label space consists of the linguistic terms of each attribute. Consider each fuzzy case as an initial fuzzy rule. We then apply the rough set technique to these fuzzy rules and get a subset of those fuzzy rules, which covers all fuzzy cases, and the cardinality of the subset is approximately minimal. The fuzzy-rough algorithm is divided into three tasks to be fulfilled [1]: (1) in search of a minimal reduct for each initial fuzzy rule, (2) in search of a family of minimal reducts for the i th ($1 \le i \le M$, where M is the number of fuzzy cases)fuzzy case such that each reduct inside of this family covers the i th fuzzy case, and (3) in search of a subset of those fuzzy rules which covers all fuzzy cases and the cardinality of the subset is minimal.

We first introduce the definitions used in the fuzzy-rough approach.

In order to transfer the fuzzy data into fuzzy rules, firstly we introduce fuzzy knowledge base concept, Table 1 is said to be a fuzzy knowledge base, where there are n rows and m attributes. $Attr_j\,(j = 1,2,\ldots,n)$. $A_{ij}\ (i = 1,2,\ldots,n;\ j = 1,2,\ldots,m)$ are all fuzzy sets defined on the same universe $U = \{1,2,..., n\}$, and it can be regarded as the value of the ith fuzzy case for the jth attribute. C_i is the classification result of the ith fuzzy example, the ith row is explained to be an initial fuzzy rule taking a form $\bigcap_{p=1}^{m} A_{ip} \Rightarrow C_i$ with true degree α_i (see **Definition 1**) and inconsistent degree β_i (see **Definition 2**).

A fuzzy knowledge base can be generated by selecting the maximal membership of each attribute over its range of non-fuzzy label values from the fuzzy data.

Table 1 Fuzzy Knowledge Base

No.	$Attr_1$	$Attr_2$	…	$Attr_m$	Class	True Degree	Inconsistency
r_1	A_{11}	A_{12}	…	A_{1m}	C_1	α_1	β_1
r_2	A_{21}	A_{22}	…	A_{2m}	C_2	α_2	β_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
r_n	A_{n1}	A_{n2}	…	A_{nm}	C_n	α_n	β_n

From the ith initial fuzzy rule, many fuzzy rules can be generated such as $\bigcap_{l=1}^{k} A_{ij_l} \Rightarrow C_i$ with a true degree and an inconsistent degree, where $\{j_1, j_2,..., j_k\} \subset \{1,2,..., m\}$. Let S= $\{Attrj_1, Attrj_2,..., Attrj_k\}$ be a subset of condition attributes($k \le m$). We denote the fuzzy rule $\bigcap_{l=1}^{k} Attr_{ij_l} \Rightarrow C_i$ with a true degree α_i and an inconsistent degree β_i, in short, by $Attr\Big|_S^i \Rightarrow C_i[\alpha_i, \beta_i]$.

Definition 1. (Yuan and Shaw [17]) The true degree of fuzzy rule $A \Rightarrow B$ is defined to be $\alpha = \sum_{u \in U} \min(u_A(u), u_B(u)) / \sum_{u \in U} u_A(u)$, where A and B are two fuzzy sets defined on the same universe U.

Definition 2. (Wang and Hong [2]) The inconsistent degree of a given fuzzy rule is defined by $|E|$ where E= $\{j \mid Attr\big|_S^i = Attr\big|_S^j, C_i \ne C_j\}$, $|E|$ denotes the number of elements of the set E.

Definition 3. (Wang and Hong [2]) For a given fuzzy rule $Attr\Big|_S^i \Rightarrow C_i[\alpha_i, \beta_i]$, an attribute $A(A \in S)$ is said to be dispensable in the fuzzy rule if $Attr\Big|_{S-\{A\}}^i \Rightarrow C_i$ has a true degree greater than or equal to σ (a given threshold) and an inconsistent degree less than or equal to β_i. Otherwise, attribute A is indispensable in the rule.

Definition 4. (Wang and Hong [2]) For a given fuzzy rule $Attr\Big|_S^i \Rightarrow C_i[\alpha_i, \beta_i]$, if all attributes in S are indispensable, this rule is called independent.

Definition 5. (Wang and Hong [2]) A subset of attributes $R(R \subset S)$ is called a reduct of the rule $Attr\Big|_{S}^{i} \Rightarrow C_i$ if $Attr\Big|_{R}^{i} \Rightarrow C_i$ is independent and has a true degree greater than or equal to s (a given threshold) and an inconsistent degree less than or equal to b_i .The set of attributes, which are indispensable in the initial rule, $Attr\Big|_{C}^{i} \Rightarrow C_i$ is called the core of the initial fuzzy rule.

Definition 6. (Wang and Hong [2]) A reduct of an initial fuzzy rule $Attr\Big|_{C}^{i} \Rightarrow C_i$, R is said to be minimal, if S is not a reduct of the initial fuzzy rule for each set S with $S \subset R$ and $S \neq R$.

Definition 7. (Wang and Hong [2]) A fuzzy rule $Attr\Big|_{S}^{i} \Rightarrow C_i[a_i , b_i]$ is said to cover a fuzzy example if the membership of attributes and the membership of classification for the example are all greater than or equal to h (a threshold).

The detailed algorithms of each task are described as follows:

Task 1 algorithm [2]: It can be divided into six steps:

Step1: for the i th initial fuzzy rule $(1 \le i \le m)$, the core K can be given by verifying whether an attribute is dispensable in the attribute set. K can be empty.

Set Γ:=1

Step 2: Take Γ attributes $Attr_1, Attr_2, ..., Attr_\Gamma$ from $C - K$

Step 3: Add $Attr_1, Attr_2, ..., Attr_\Gamma$ to K.

Step 4: compute the true degree and the inconsistent degree of the fuzzy rule $Attr\Big|_{K}^{i} \Rightarrow C_i$,

Step 5: if K is a reduct then exit successfully, else new Γ attributes $Attr_1, Attr_2, ..., Attr_\Gamma$ are taken from $C - K$, goto Step 3.

Step 6: if all combinations of elements of $C - K$ have been used and a reduct does not appear, $\Gamma := \Gamma + 1$, goto step 2.

Task 2 algorithm [2]: For each i $(1 \le i \le m)$, R_i , a subset of R= $\{r_1, r_2, ..., r_m\}$, where r_i is the minimal reduct of the i th initial rule, can be determined by checking whether the rule covers the example f_i :

$$R_i = \{r_j | r_j \in R, r_j \text{ covers } f_i\} (i = 1,2,...m)$$

Task 3algorithm [2]:

Take $\Omega = \{R_{1,} R_{2,}..., R_m\}$, R_i from the second task. The initial value of R^* is supposed to be an empty set. Repeat the following three steps:

Step1: for each $r \in R$, compute the number of times that r appears in the family Ω.

Step2: select r^*, such that the number times of r^* appears in the family Ω is maximum.

Step3: for i= $1,2,...m$, remove R_i from Ω if $r^* \in R_i$ and replace R^* with $\{r^*\} \cup R^*$ until Ω becomes empty.

R^* is then the fuzzy rule we need. For each case of a considered cluster, a set of adaptation rules is generated.

With respect to the generated adaptation rules, we need a reasoning mechanism to predict the amount of adjustment for the solution of non-representative cases. We propose our fuzzy reasoning mechanism as in [1].

As a result of this phase, for each case of a considered cluster, a set of adaptation rules (fuzzy production rules) is generated, and a reasoning mechanism for this set of fuzzy rules is given.

2.4 Selecting Representative Cases

This phase aims to select representative cases from each cluster according to the adaptation rules obtained in phase three. Our selection strategy uses the method in [1], which is based on a $\boldsymbol{e}$-coverage concept. Instead of the deletion, [1] proposes a selection strategy which makes use of Smyth's proposed concepts of coverage and reachability with some changes (called $\boldsymbol{e}$-coverage and $\boldsymbol{e}$-reachability respectively).
Let L be a cluster in which each case e is accompanied by a set of adaptation rules AR(e), $\boldsymbol{e}$ be a small positive number, and $e_p = (x_{p1}, x_{p2}, \cdots, x_{pn}, v_p)$ and $e_q = (x_{q1}, x_{q2}, \cdots, x_{pn}, v_q)$ be two cases in the cluster L. According to the reasoning mechanism established in phase 3, an adjustment amount Δ of the solution for case e_q can be obtained by matching $(x_{q1} - x_{p1}, \cdots, x_{qn} - x_{pn})$ against $AR(e_p)$. If $v_q + \Delta \in (v_p - e, v_p + e)$, then e_p is said to $\boldsymbol{e}$-cover with e_q. The $\boldsymbol{e}$-coverage and $\boldsymbol{e}$-reachability of the case e_p are defined by

$$\text{Coverage}(e_p) = \{e \,|\, e \in L, e \text{ is } \boldsymbol{e}\text{-covered by } e_p\} \tag{6}$$

and

$$\text{Reachability}(e_p) = \{e \mid e \in L, e \ \ \boldsymbol{e}\text{-covers with } e_p\} \text{ respectively} \tag{7}$$

3 Experimental Analysis

This section presents the experimental analysis of our approach on a real-world problem, i.e. the rice taste (RT) problem. The RT data consist of five inputs and a single output whose values are associated with subjective evaluations of the flavor, appearance, taste, stickiness, toughness and overall evaluation of 105 different kinds of rice (Table 2 shows some typical records).

Table 2. Rice taste datat sizes of headings

Flavor	Appearance	Taste	Stickiness	Toughness	Overall Evaluation
0.699	1.543	1.76	1.944	-0.875	1.706
-0.593	-0.898	-0.883	-0.647	0.323	-1.235
0.158	0.163	0.03	0.359	-0.128	0.135

After applying the learning feature weights algorithm mentioned in section 2.1 to these cases, the feature weight results shown in Table 3 are obtained (learning iterations = 10000 cycles).

Table 3. Feature weights of the problem features

Flavor	Appearance	Taste	Stickiness	Toughness	Overall evaluation
0.02	0.03	0.54	0.03	0.04	2.68

Phase two is the same as the one in [1], readers can refer to that paper. As a result of this phase, the cases are partitioned into 14 classes. Some of these classes are shown in Table 4. We label classes with less than 10 records as Odd classes and the rest Not-odd classes. The learning of fuzzy adaptation rules is carried out on the Not-odd classes.

Table 4. Clusters of the rice case-base

Cluster no.	Number of cases	Odd or Not-Odd class
1	34	Not-Odd
2	30	Not-Odd
3	13	Not-Odd
4	7	Odd
5	5	Odd
6	1	Odd

For the first five problem features, i.e. Flavor, Appearance, Taste, Stickiness and Toughness (see Table 1), we fuzzify them into three linguistic variables: small, medium, big. For the solution feature, i.e. overall evaluation, we fuzzify it into five linguistic variables, i.e. Negative Big, Negative Small, Zero, Positive Small, Positive Big.

The general form of a fuzzy adaptation rule generated from the fuzzy decision tree is as follows:

IF the change of X1 is [Small | Medium | Big]
[AND the change of X2 is [Small | Medium | Big]
[AND the change of X3 is [Small | Medium| Big]]]
THEN the change of overall evaluation is [Negative Big | Negative Small | Zero | Positive Small | Positive Big].
Where X = {Flavor, Appearance, Taste, Stickiness, Toughness}.

For example, in cluster 3, which consists of 13 cases, one of the adaptation rules is:
Rule: IF the change of Flavor is medium and the change of Appearance is medium, THEN the change of overall evaluation is Positive Small.

Table 5. Reachability and coverage of each case in cluster 3 of the rice case-base

Case number (x)	Number of cases which are covered by case(x)	The actual cases which are covered by case(x)	No. of adaptation rules
0	2	3,9	4
1	4	2,3,6,9	4
2	1	1	4
3	0		5
4	3	5,7,10	7
5	1	4	5
6	5	1,2,3,9,10	5
7	4	4,8,11,12	5
8	3	5,7,12	7
9	2	1,11	3
10	1	4	5
11	0		5
12	4	4,5,8,10	4

According to the case selecting strategy defined in section 2.4, we select cases {6,7,0,5,10} as the representative cases in this cluster 3 (see Table 5, the specific e=0.05). As a result of this selection, a total of 24 fuzzy adaptation rules are also selected (i.e. each case has five adaptation rules on average).

Table 6. Selection of representative cases in all the Not-Odd clusters

Cluster	Number	No. of	No. of deleted	The average

No.	of cases	representative cases	cases	relative error of the deleted cases
1	34	9	25	2.39%
2	30	7	23	7.76%
3	13	5	8	5.62%
	Total 77	Total 21	Total 56	Overall Average 5.14%

After applying the case selection strategy to each not-odd cluster, 56 cases are deleted (see Table 6), in other words, the number of cases in rice taste case base can be reduced by 53%.

Comparing with that of the fuzzy decision tree method [19], the result generated by fuzzy rough approach is quite positive. There are 56 cases are deleted by using fuzzy rough approach while only 39 cases are deleted by using fuzzy decision tree method. And the number of adaptation rules for each case generated by the fuzzy-rough method is much less than that of the fuzzy decision tree method (listed in the second column of Table 7).
In order to evaluate the overall problem solving ability, we apply those deleted cases as new coming cases to the smaller case base and its associated adaptation rules generated by our maintenance approach for solving, the results shows that fuzzy rough approach is also much better than fuzzy decision tree method. Table 8 demonstrates the comparison results of those two methods.

Table 7. Comparison the number of adaptation rules between the fuzzy decision tree and fuzzy-rough method

	Average Number of Adaptation Rules	Generate Decision Tree
Fuzzy Decision Tree	11.8	Yes
Fuzzy-Rough Method	8.7	No

Table 8. Average error after deletion

Cluster No	The average relative error of the deleted cases by the fuzzy-rough method	The average relative error of the deleted cases by the fuzzy decision tree method
1	2.39%	8.76%
2	7.76%	14.86%
3	5.62%	3.48%
Average error	5.41%	10.26%

So the overall selection result based on the adaptation rules generated by fuzzy-rough method is better than those based on the rules generated by the fuzzy decision tree. We can therefore say the overall performance of the fuzzy-rough approach is better than that of the fuzzy decision tree induction method.

4 Summary and future works

In this paper, we have developed a fuzzy-rough approach to maintaining case-based reasoning systems and compared the results with on that used fuzzy decision tree induction [1]. The main idea is to mine the adaptation knowledge by the fuzzy-rough approach, i.e., taking the fuzzy cases as fuzzy rules, then applying the rough set technique to those fuzzy rules, and generating a group of adaptation rules. A case selection strategy is then implemented based on these adaptation rules, and finally the original case library is replaced with a small case library plus adaptation knowledge. This adaptation knowledge plays the role of complementing the reduction of cases. The experimental analysis of our method showed promising results. Future work includes(1) a large scale testing of our methodology using different case-bases, (2) the refining of the fuzzy-rough algorithms, (3) a comprehensive analysis of the complexity of the case base maintenance and reasoning algorithm in time and space, and (4) future comparison with other methods, such as fuzzy decision tree, C4.5, genetic algorithm and so on. We are also very interested in building a framework of the case base maintenance, including a reasoning scheme, retaining new cases, and on-line or periodic updating.

5 Acknowledgement

We would like to express our thanks to the three anonymous reviewers who have given us a lot of valuable comments on this work.

This project is supported by the Hong Kong Polytechnic University MPhil research studentship grant G-V957 and small research grant G-T144

References

[1] Shiu C.K. , Sun. C. H., Wang X.Z. and Yeung S., "Transferring Case Knowledge to Adaptation Knowledge: An Approach for Case Base Maintenance", *Computational Intelligence*, Volume 17, Number 2, 2001.

[2] Wang X.Z., Hong, J.R., "Learning optimization in simplifying fuzzy rules", *Fuzzy Sets and Systems* 106, 1999.

[3] Kitano, H. and Shimazu, H. "The experience sharing architecture: A case study in corporate-wide case-based software quality control", In *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Edited by Leake, D. Menlo Park, CA, AAAI Press, pp. 235-268, 1996.

[4] Cheetham, W. and Graf, J., "Case-based reasoning in color matching", In *Proceedings of the Second International Conference on Case-Based Reasoning*, ICCBR-97, pp. 1-12, 1997.

[5] Deangdej, J., Lukose, D., Tshui, E., Beinat, P. and Prophet, L., " Dynamically creating indices for two million cases: A real world problem", In *Proceedings of the 3rd European Workshop of Case-Based Reasoning*, EWCBR-96, pp. 105-119,1996.

[6] Leake, D.B. and Wilson, D.C. "Categorizing Case-Base Maintenance: Dimensions and Directions", *In Proceedings of the 4th European Workshop of Case-Based Reasoning*, EWCBR-98, pp. 196-207,1998

[7] Smyth, B. and Keane, M.T., "Remembering to Forget: A Competence-Preserving Case Deletion Policy for Case-based Reasoning systems", In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, IJCAI-95, pp. 377-382, 1995.

[8] Smyth, B. and Mckenna, E. , "Modeling the Competence of Case-bases", In *Proceedings of the 4th European Workshop of Case-Based Reasoning*, EWCBR-98, pp. 23-25, 1998.

[9] Anand, S.S., Patterson, D., Hughes, J.G. and Bell D.A., "Discovering Case Knowledge using Data Mining", In *Proceedings of the 2nd Pacific Asia Conference on Knowledge Discovery and Data Mining* : Current Issues and New Applications, PAKDD-98, pp. 25-35, 1998.

[10] Richter, M. M. , "The Knowledge Contained in Similarity Measures", *Invited Talk at ICCBR-95*. http://wwwagr.informatil.uni-kl.de/~lsa/CBR/Richtericcbr95remarks.html.

[11] Richter, M. M. , "Chapter one: Introduction", In *Case-Based Reasoning Technology: From Foundations to Applications.* Edited by Lena, M., Bartsch-Sporl, B., Burkhard, H.D. and Wess, S., Springer-Verlag, Berlin, Germany, pp. 1-15, 1998.

[12] Zadeh L.A., "Fuzzy Sets", I*nformation and Control*, Vol.8, 1965.

[13] Drwal G., "Rough, and Fuzzy Rough Classification Methods Implemented in RClass System", ?,?.

[14] Pawlak Z., "Rough Set", *International Journal of Computer and Information Sciences*, 1982.

[15] Bezdek, J. C. , "Pattern recognition with fuzzy objective function algorithms," Plenum, NewYork, 1981

[16] Kohonen, T., "Self-Organization and Associate Memory", Springer, Berlin, 1988.

[17] Yuan. Y, Shaw M.J., "Induction of fuzzy decision trees", *Fuzzy Sets and Systems 69*, pp.125-139, 1995.

[18] Nozaki, K., Ishibuchi, H. and Tanaka, H, "A simple but powerful heuristic method for generating fuzzy rules from numerical data," in *Fuzzy Sets and Systems*, FSS 86, pp.251-270, 1997.

[19] Shiu C.K., Sun. C. H., Wang X.Z. and Yeung S., "Maintaining Case-Based Reasoning Systems Using Fuzzy Decision Trees", In *Proceedings of the 5rd European Workshop of Case-Based Reasoning*, EWCBR-00, pp285-296, 2000.