

Maintaining Case-Based Reasoning Systems Using Fuzzy Decision Trees¹

Simon Chi Keung Shiu, Cai Hung Sun, Xi Zhao Wang and Daniel So Yeung

Department of Computing, Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong

{csckshiu, cschsun, csxzwang, csdaniel}@comp.polyu.edu.hk

Abstract. This paper proposes a methodology of maintaining Case Based Reasoning (CBR) systems by using fuzzy decision tree induction - a machine learning technique. 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 are generated by fuzzy decision trees. Firstly, an approach to learning feature weights 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. Finally, a selection strategy based on the concepts of ε -coverage and ε -reachability is used to select representative cases. The effectiveness of the method is demonstrated experimentally using two sets of testing data.

1 Introduction

Case-base maintenance is the process of refining a CBR system's case-base to improve the system's performance, i.e., 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 [9]. In the past, researchers have attempted to address various aspects of the case-base maintenance problem. To provide maintenance support at the case level, Smyth and Keane [12] suggested a competence preserving deletion approach. Competence (or coverage) is the range of target problems that a given system can solve, and is also a fundamental evaluation criterion of CBR system performance. Barry Smyth et al. [13] 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. [1] proposed to use data mining techniques in a novel role of a back-end technology for case-based reasoning (CBR) systems, i.e., the acquisition of cases and discovery of adaptation knowledge.

¹ This project is supported by a HK PolyU grant PA25

Hanney and Keane [5] presented an inductive learning algorithm to extract adaptation knowledge from the cases in the case-base. Their algorithm builds pairs of cases and uses the feature differences of these case pairs to build adaptation rules which are very useful in case-base maintenance. The approach of Hanney et al. is based on the assumption that the differences occurred between cases in the case-base are representative of the differences that will occur between future problems and the case-base.

Although many researchers, as aforementioned, have contributed to solving the case-base maintenance problems, there is still no systematic method available. In this paper, we propose a systematic methodology for case-base maintenance, which integrates identifying salient features, distinguishing different concepts, learning adaptation knowledge, computing case competence and selecting seed cases together into a framework of case-base maintenance. The methodology is mainly based on a process that a large case library is transformed to a small case library together with a group of adaptation rules, which are generated by fuzzy decision trees. The details of maintaining a case-base from scratch, as proposed in this paper, consists of four steps. 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. Finally, a selection strategy based on the concepts of ε -coverage and ε -reachability is used to select representative cases. The effectiveness of the method is demonstrated experimentally on the Rice Taste Data and the Boston Housing Data. The result shows that the testing case-base size can be reduced by 39% and 43% respectively if we complement the remaining cases by adaptation rules discovered using our approach. The overall competence of the smaller case-bases is 90% of the original ones.

2 Methodology of Maintaining a Case Library

Throughout this section, we consider a case library in which all features are supposed to take numeric real values. It should be noted that the real-valued features discussed here could be, without difficulties, extended to the features, which take values in a normed vector space. We first introduce a weighted distance metric and a similarity measure used in the following.

Let $CL = \{e_1, e_2, \dots, e_N\}$ denote a case library. Each case in the library can be identified by an index of corresponding features. In addition each case has an associated action. More formally we use a collection of features $\{F_j \ (j = 1, \dots, n)\}$ to represent the cases and a variable V to denote the action. The i -th case e_i in the library can be represented as a $n+1$ -dimensional vector, i.e. $e_i = (x_{i1}, x_{i2}, \dots, x_{in}, v_i)$ where x_{ij} corresponds to the value of feature F_j ($1 \leq j \leq n$) and v_i corresponds to the action ($i = 1, \dots, N$).

Suppose that for each j ($1 \leq j \leq n$) a weight w_j ($w_j \in [0,1]$) has been assigned to the j -th feature to indicate the importance of the feature. Then, for any pair of cases e_p and e_q in the library, a weighted distance metric can be defined as

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

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

By using the weighted distance defined in equation (1), a similarity measure between two cases, $SM_{pq}^{(w)}$, can be defined as follows:

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

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

After introducing the weighted distance metric and the similarity measure, our methodology, which consists of four major phases, will be described in the following four sub-sections.

2.1 Phase One - Learning Feature Weights

In this section, a feature evaluation function is defined. The smaller is the evaluation value, the better are 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 decent technique. We formulate this optimization problem as follows.

For a given collection of feature weights w_j ($w_j \in [0,1], j = 1, \dots, n$) 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 $SM_{pq}^{(w)}$. When all weights take value 1, $d_{pq}^{(w)}$ and $SM_{pq}^{(w)}$ will degenerate the Euclidean distance $d_{pq}^{(1)}$ and $SM_{pq}^{(1)}$. A feature evaluation index E is defined as

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

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

Noting that the feature evaluation function E(w) will gradually become zero when $SM_{pq}^{(w)} \rightarrow 0$ or 1, we hope finding a collection of weights such that the feature evaluation function attains its minimum.

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

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

for $j = 1, \dots, n$, where η is the learning rate.

For the computation of $\frac{\partial E}{\partial w_j}$, the following expressions are used:

$$\frac{\partial E(w)}{\partial w_j} = \frac{2 * [\sum_{pq} \sum_{(q < p)} \left(1 - 2 \cdot SM_{pq}^{(1)} \right) \cdot \frac{\partial SM_{pq}^{(w)}}{\partial d_{pq}^{(w)}} \cdot \frac{\partial d_{pq}^{(w)}}{\partial w_j}]}{N * (N - 1)} \tag{5}$$

$$\frac{\partial SM_{pq}^{(w)}}{\partial d_{pq}^{(w)}} = \frac{-\alpha}{(1 + \alpha \cdot d_{pq}^{(w)})^2} \tag{6}$$

$$\frac{\partial d_{pq}^{(w)}}{\partial w_j} = \tag{7}$$

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.

After training, the function $E(w)$ attains a local minimum. We expect that, in average, the similarity values $\{SM_{pq}^{(w)}, p = 1, \dots, N, q < p\}$ with trained weights are closer to 0 or 1 than that without trained weights such as $\{SM_{pq}^{(1)}, p = 1, \dots, N, q < p\}$.

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 [2] and Kohonen' self-organizing network [8] can be used to partition the case library. However, this paper adopts a typical approach of clustering, i.e., similarity matrix [4] which uses only the information of similarity between cases This approach first transform the similarity matrix to a equivalent matrix and then considers the cases being equivalent each other as one cluster. The procedure is as follows.

- Step 1. Give a significant level (threshold) $\beta \in (0,1]$
- Step 2. Determine the similarity matrix $SM = \left(SM_{pq}^{(w)} \right)$ according to equation (2) and (1).
- Step 3. Compute $SM1 = SM \circ SM = \left(s_{pq} \right)$ where

$$s_{pq} = \max_k \left(\min(SM_{pk}^{(w)}, SM_{kq}^{(w)}) \right).$$
- Step 4. If $SM1 \subset SM$ then go to step 5, else replace SM with SM1 and go to step 3.
- Step 5. Determine several clusters based on the rule "case p and case q belong to the same cluster if and only if $s_{pq} \geq \beta$."

2.3 Phase Three - Mining Adaptation Rules by Fuzzy Decision Trees

In this paper we follow the hypothesis that each cluster contain cases that are more representative of others and a non-representative case in the cluster is considered as a perturbation of certain representative case. The perturbation is handled by a group of adaptation rules. In details, let p be a representative case and q be a non-representative case in some cluster, we expect that the solution of q can be obtained by an appropriate adaptation (adjustment) of the solution of p according to some adaptation rules. For example, a representative case and a non-representative case are respectively $p = (1, 2, 3, 4)$ and $q = (0.9, 2, 3, 4.01)$ in which the first three

components are feature-values and the last component is the solution, an adaptation rule is “IF the difference between the feature values is negatively small THEN the adjustment of the solution is very positively small”, then q’s solution can be considered to be obtained by p’s solution with an adjustment based on the adaptation rule.

Fuzzy decision tree generation technique such as ID3, which is based on minimum information entropy to select expanded attributes, was proposed by Quinlan in 1986 [11]. Subsequently, the fuzzy version of ID3 based on minimum fuzzy entropy was suggested by several authors [6,7,14]. The second approach has many advantages over the first such as robustness and comprehensibility. That motivates the choice of this technique.

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

$$f_i = e_i - e_k = (x_{i1} - x_{k1}, x_{i2} - x_{k2}, \dots, x_{in} - x_{kn}, v_i - v_k) = \{y_{i1}, y_{i2}, \dots, y_{in}, u_i\}$$

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

Consider a problem of learning from examples in which there are $n+1$ numerical attributes, $\{A^{(1)}, A^{(2)}, \dots, A^{(n)}, A^{(n+1)}\}$ ($A^{(n+1)}$ is the classification attribute). Then $\{f_i \mid i = 1, 2, \dots, m\}$ 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 shown in Figure 1. For each attribute (the j -th attribute $A^{(k)}, 1 \leq k \leq n+1$) with the attribute-values $Range(A^{(k)}) = \{y_{1k}, y_{2k}, \dots, y_{mk}\}$, the two parameters in Figure 1, a and b, are defined by

$$a = \sum_{y \in N} y / Card(N) \quad \text{and} \quad b = \sum_{y \in P} y / Card(P) \tag{8}$$

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

After this fuzzification, each example can be regarded as a $5 \times (n + 1)$ -dimensional vector of membership degree. By putting the m vectors together, a matrix with m rows and $5 \times (n + 1)$ columns is formed. According to this matrix, we propose our fuzzy decision tree generation procedure – fuzzy ID3. In comparison with the existing versions of fuzzy ID3, our proposal is founded on such a viewpoint that each linguistic term of attributes and each node in the tree are considered to be fuzzy sets defined on the example-label space $\{1, 2, \dots, m\}$.

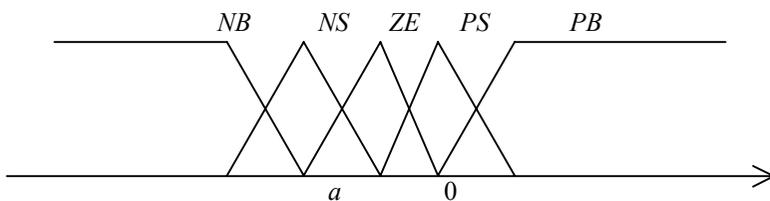


Fig. 1. Five membership functions

After generating the fuzzy decision tree by using fuzzy ID3 algorithm, a set of adaptation rules can be extracted from the tree. The extraction is straightforward, i.e., each path from the root to a leaf is converted into an adaptation rule (fuzzy production rule). Thus the number of leaves is just the number of adaptation rules.

2.4 Phase Four - 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 is based on a ϵ -coverage concept. Instead of the deletion, this paper proposes a selection strategy which makes use of Smyth proposed concepts of coverage and reachability with some changes (called ϵ -coverage and ϵ -reachability respectively).

Let L be a cluster in which each case e is accompanied with a set of adaptation rules $AR(e)$, ϵ be a small positive number, $e_p = (x_{p1}, x_{p2}, \dots, x_{pn}, v_p)$ and $e_q = (x_{q1}, x_{q2}, \dots, x_{qn}, v_q)$ be two cases in the cluster L . According to the reasoning mechanism established in phase 3, an adjustment amount Δ of the solution for the case e_q can be obtained by matching $(x_{q1} - x_{p1}, \dots, x_{qn} - x_{pn})$ against $AR(e_p)$. If $v_q + \Delta \in (v_p - \epsilon, v_p + \epsilon)$, then e_p is said to ϵ -cover with e_q . The ϵ -coverage and ϵ -reachability of the case e_p are defined by

$$\text{Coverage}(e_p) = \{e \mid e \in L, e \text{ is } \epsilon\text{-covered by } e_p\} \tag{9}$$

and

$$\text{Reachability}(e_p) = \{e \mid e \in L, e \text{ } \epsilon\text{-covers with } e_p\} \text{ respectively} \tag{10}$$

respectively.

The ε -coverage of a case e represents the generalization capability of this case. The bigger is the number of cases in the ε -coverage, the more representative is the selected case e . As a kind of rule extraction, it is a commonsense application of Occam's Razor. On the other hand, the ε -reachability of a case e represents the degree to which e can be replaced by another case. The smaller is the number of cases in the ε -reachability, the more important is the selected case e . As an index of evaluation of selected cases, to a great extent, it reflects the difference between rule-based and case-based approaches. Our proposed selection procedure integrates these two approaches together.

3 Experimental Analysis

This section presents the experimental analysis of our methodology on two real-world problems that are taken from [10] and [3], i.e. the rice taste (RT) and the Boston housing (BH) problem. The RT data consists 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 one show some typical records). The BH data consists of thirteen inputs and a single output whose values are taken from 506 housing records in suburbs of Boston. The experiments were carried out in a Pentium III 500MHz machine and the programs were all written in Microsoft Visual C++.

3.1 The Rice Taste Data

Table 1. Rice Taste Datat sizes of headings

Favor	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 weight algorithm mentioned in section 2.1 to this cases, the feature weight results shown in Table 2 were obtained (Learning iterations = 10,000 cycles, Alpha = 0.46 and Learning rate = 0.05).

Table 2. Feature weights

Favor	Appearance	Taste	Stickiness	Toughness	Overall Evaluation
0.02	0.03	0.54	0.03	0.04	2.68

The clustering of cases is carried out both with and without the feature weight information. The results are shown in Table 3. When feature weight information is used to guide the clustering of cases, much better results were obtained, i.e. the number of clusters are between 9 to 26 and there are at least 70% of the cases which fall in the first 5 largest clusters. The user can specify a particular significant level (β) for further learning of adaptation rules.

Table 3. Clustering of cases with and without feature weights

Significant level (β)	Without feature weights			With feature weights		
	No. of clusters	No. of clusters contains one case record	of Cases in the first 5 largest clusters (%)	No. of clusters	No. of clusters contains one case record	cases in the first 5 largest clusters (%)
0.86	99	94	10.48	9	4	96.19
0.87	100	96	9.52	14	5	84.76
0.88	102	100	7.62	15	6	83.81
0.89	102	100	7.62	20	9	79.05
0.90	104	103	5.71	26	13	70.48

In our experiment, we have chosen $\beta = 0.87$ as the significant level. As a result, the cases are partitioned into 14 clusters.

In the mining of fuzzy adaptation rules, the first five problem features, i.e. Favor, Appearance, Taste, Stickiness and Toughness, were fuzzified into three linguistic variables, Small, Middle, Big. The solution feature, i.e. Overall Evaluation, was fuzzified 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 follow:

IF the difference of X1 is [Small | Middle | Big]
 [AND the difference of X2 is [Small | Middle | Big]
 [AND the difference of X3 is [Small | Middle | Big]]]

THEN the difference of overall evaluation is [Negative Big | Negative Small | Zero Positive Small | Positive Big].

Where $X = \{\text{Favor, Appearance, Taste, Stickiness, Toughness}\}$ and the maximum number of antecedents of each fuzzy rule is limited to three in this experiment. For example, in cluster 5, which consists of 13 cases, one of the adaptation rules is:

Table 4. Case coverage

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
1	4	3,4,9,11	6
2	6	3,6,9,10,11,12	11
3	2	7,9	11
4	4	6,9,11,12	10
5	5	1,2,6,9,11	10
6	3	1,7,11	12
7	8	1,3,5,6,9,10,11,12	16
8	3	2,9,11	14
9	1	8	15
10	6	4,6,7,9,11,12	10
11	4	3,6,9,12	14
12	3	4,6,11	13
13	2	1,11	11

Rule1: IF the difference of Favor is small
 AND the difference of Toughness is small
 AND the difference of Taste is small
 THEN the difference of overall evaluation is Zero.
 The rule's confidence is 0.92

According to the selecting case policy defined in section 2.4, we select cases {7,2,4,8,13} as the representative cases in this cluster 5 (see Table 4). The overall selection results of cluster 3, 4, and 5 are shown in Table 5 where for cluster 3, $\epsilon=0.15$ (i.e. the absolute error of expected value is not greater than 0.15), for cluster 4, $\epsilon=0.1$ and for cluster 5, $\epsilon=0.15$.

Table 5. Overall selection result of cluster 3, 4 and 5

Cluster No	Number of cases	No. of Representative cases	No. of deleted cases
3	34	18	16
4	30	13	17
5	13	5	8
Total	77	36	41

Therefore, the total number of deleted cases is 41. After removing these deleted cases, we use them back to test the competence of the smaller case-base with the adaptation rules. The average relative error of the solution in each cluster is shown as in Table 6. We defined relative error = (real value – expected value)/real value *100%.

The result shows that the testing case-base size can be reduced by 39% if we complement the remaining cases by adaptation rules discovered using our approach. The overall competence of the smaller case-base is 90% of the original one.

Table 6. Average error after deletion

Cluster No	The average relative error of the deleted cases in this cluster
3	8.76%
4	14.86%
5	3.48%
Average error	10.26%

3.2 The Boston Housing Data

Using the similar approach as described in section 3.1, the Boston housing data are successfully partitioned into 93 clusters, with 13 clusters having at least 9 cases and the biggest cluster consists of 47 cases. However there are also 56 very small clusters with only one case each. Learning of fuzzy adaptation rules are carried out to those 37 clusters, (i.e. clusters having more than one case), and a selection policy is adopted with $\epsilon=0.10$. The result shows that the case-base size can be reduced by 43% and the overall competence of the smaller case-base is 90% of the original one.

4 Summary and Future Works

In this paper, we have developed a methodology of maintaining Case Based Reasoning systems by using fuzzy decision tree induction. The main idea is to transform a large case library to a small case library together with a group of adaptation rules, which are generated by fuzzy decision trees. These adaptation rules play the role of complementing the reduction of cases. The experimental analysis of our method showed promising results. Future work include (1) a large scale testing of our methodology using different case-bases, (2) different selection polices will be developed based on ideas such as subsumption, conflicting and ambiguity which exist among cases and (3) A reasoning method will be developed for effective problem solving using this hybrid knowledge representation scheme. This involves the development of similarity computation of new cases versus the fuzzy rules/small case-library knowledge structure in the case base. This reasoning method should also have the ability to adapt the retrieved solution to suit the new case.

References

1. Anand, S. S., Patterson, D., Hughes, J. G. and Bell D. A., "Discovering Case Knowledge using Data Mining," in Second Pacific Asia Conference, PAKDD-98, Australia, pp. 25-35.
2. Bezdek, J. C. , "Pattern recognition with fuzzy objective function algorithms," Plenum, NewYork, 1981.
3. Boston Housing Data, UCI ML Repository, www.ics.uci.edu/~mllearn/MLRepository.html
4. Fu, G., "An algorithm for computing the transitive closure of a similarity matrix," Fuzzy Sets and Systems, vol. 51, pp. 189-194, 1992.
5. Hanney, K. and Keane, M. T., "Learning Adaptation Rules from a Case-Base," In Proc. Advances in Case-Based Reasoning, 3rd European Workshop, EWCBR-96, pp179-192.
6. Ichihashi, H. et al., "Neuro-fuzzy ID3, " in Fuzzy Sets and Systems vol. 81, pp.157-167, 1996.
7. Jeng, B. et al., "FILM: a fuzzy inductive learning method for automated knowledge acquisition," in Decision Support Systems, vol. 21, pp. 61-73, 1997.
8. Kohonen, T., "Self-Organization and Associate Memory", Springer, Berlin, 1988.
9. Leake, D. B. and Wilson, D. C., "Categorizing Case-Base Maintenance: Dimensions and Directions," Advances in Case-Based Reasoning, 4th European Workshop, EWCBR-98 , pp. 196-207.
10. 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(1997), pp. 251-270.
11. Quinlan, J. R., "Induction of decision trees", Machine Learning, vol. 1, pp.81-106, 1986.

12. Smyth, B. and Keane, M. T. , “Remembering to Forget: A Competence-Preserving Case Deletion Policy for Case-based Reasoning systems”, Proceedings of the fourteenth International Joint Conference on Artificial Intelligence, IJCAI-95, pp. 377-382.
13. Smyth, B. and Mckenna, E., “Modeling the Competence of Case-bases”, Advances in Case-Based Reasoning, 4th European Workshop, EWCBR-98, pp. 23-25.
14. Umanol, M. et al., “Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems,” in IEEE International Conference on Fuzzy Systems, (26-29 June 1994), pp. 2113-2118.