

# Fuzzy Dissimilarity Learning in Case-Based Reasoning

Ning Xiong

School of Innovation, Design and Engineering

Mälardalen University

P.O. Box 883, SE-72123 Västerås

Sweden

ning.xiong@mdh.se <http://www.idt.mdh.se/personal/nxg01/>

*Abstract:* - Case-based reasoning (CBR) attempts to solve new problems by using previous experiences. However traditional CBR systems are restricted by the similarity requirement, i.e., the availability of similar cases to new problems. This paper proposes a novel CBR approach that exploits dissimilarity information in problem solving. A fuzzy dissimilarity model consisting of fuzzy rules has been developed for assessing dissimilarity between cases. Further, it is indicated that the construction of fuzzy dissimilarity rules can be realized by learning from the case library. Empirical studies have demonstrated that fuzzy dissimilarity models can be built upon a small case library while still yielding competent performance of the CBR system.

*Key-Words:* - case-based reasoning; similarity; dissimilarity; case library; fuzzy rules; fuzzy reasoning

## 1 Introduction

Case-based reasoning (CBR) presents an important cognitive methodology in Artificial Intelligence, which aims to use previous experiences to solve new problems [1]. A fundamental principle for conventional CBR methods is the hypothesis that similar problems have similar solutions. Hence a CBR system usually first retrieves cases in the case base that are similar to a query problem and then refines the solutions of the retrieved cases to tackle the new situation at hand.

However, the concept of similarity remains an unclear issue. The meaning of “similarity” may be subjective and vary from situation to situation. The metric of similarity is intrinsically defined on the space of problems yet there is no natural link between the problem space and the solution space to warrant that the most similar case is also the most relevant to the target problem. This implies that the basic CBR hypothesis that similar problems have similar solutions does not hold in all circumstances. Tackling situations when similar problems don’t have similar solutions is a crucial challenge for CBR research [2].

This paper investigates a new CBR approach that relies on dissimilarity information for problem solving. We consider two cases to be dissimilar as long as they have distinct or “remote” solutions. The dissimilarity model will be established to enable identification of cases from the case library that are

dissimilar to a query problem. A dissimilar case provides counter-evidence to some extent, suggesting the inappropriateness of using its solution for solving the query problem. It follows that the final decision from CBR will be the candidate solution that has accumulated the least amount of counter-evidence.

Further the model of dissimilarity is represented as a set of fuzzy linguistic rules. We believe that fuzzy if-then rules present a powerful and flexible means to represent the rich domain knowledge for case evaluation. Fuzzy rule-based reasoning can be performed to predict whether and to which extent a case from the library is dissimilar to the problem in query. The construction of fuzzy dissimilarity rules can be realized by learning from the case library as a valuable resource. Our empirical studies have demonstrated that fuzzy dissimilarity models can be built upon a small case library while still yielding competent performance of the CBR system.

The remaining of the paper is organized as follows: Section 2 surveys the related works. Section 3 outlines the new dissimilarity-based CBR approach proposed in the paper. The fuzzy dissimilarity model for case evaluation is presented in section 4. Then, in section 5, we discuss the issue of how to learn these fuzzy dissimilarity rules from the case library. In section 6, we illustrate experimental results for evaluation of the proposed method. Finally, section 7 gives the conclusion.

## 2 Related Works

Similarity evaluation is a key part for conventional CBR systems. So far the main stream of the works for construction of similarity models has been focused on feature weighting [2, 3]. Features are assigned with different weights in accordance with their importance, and the global similarity metric is defined as a weighted sum of the local matching values in single attributes. Different approaches of interest have been proposed for identifying such weights automatically. Incremental learning attempts to modify feature weights according to success/failure feedback of retrieval results [4]. The probability of ranking principle was utilized in [5] for the assignment of weight values to features. Case-ranking information was utilized in [6, 7] for weight adaptation towards similarity degrees of retrieved cases consistent with a desired order. Accuracy improvement represents another way for adapting the set of weights as discussed in [8] and [9]. Nevertheless, no matter how the values of weights are derived, the capability of these similarity learning methods is inherently constrained by weighted combination of the local matching degrees. This limitation in the structure of similarity makes it hard to represent more general knowledge and criteria for case assessment.

A new similarity model without feature weighting was proposed in [10] and [11] as an effort to seek more powerful representation of knowledge for case retrieval. The idea was to encode the information about feature importance into local compatibility measures such that feature weighting is no longer needed. Later, in [12], it was analyzed and demonstrated that the parameters of such compatibility measures can be learned from the case library in favor of coherent matching, i.e. to maximize the supportive evidence while minimize the amount of inconsistency derived from pairwise matching of cases from the case base.

The integration of fuzzy systems with CBR methodology has been an interesting topic addressed by some other researchers. Yager [13] identified that there was a close connection between fuzzy system modeling and case based reasoning. In [14] the central notion of similarity in CBR was treated as a fuzzy relation and fuzzy operations were applied as a tool for building composite similarity measures. Dubois and Prade [15] formalized the fundamental hypothesis of CBR in the context of fuzzy rules. They established a formal framework in which case-based inference can be implemented as a special type of fuzzy set-based approximate reasoning. On the other hand, CBR can be used to assist fuzzy

system modeling as well. In [16] and [17] it was demonstrated that CBR could be exploited as feature selection criterion for building complex process models and fuzzy systems.

## 3 Dissimilarity-Based CBR

In this paper we propose a new CBR approach that relies on dissimilarity information, as is depicted in Fig. 1. It starts with comparison of the query problem with known cases in the case library. A properly defined dissimilarity function has to be employed at this stage. As the evaluated dissimilarity values reflect the strength of inappropriateness of solutions of the known cases to solve the new problem, they offer important information to be utilized in the next step of solution filtering to find out the least impossible choice as the final decision to solve the new problem.

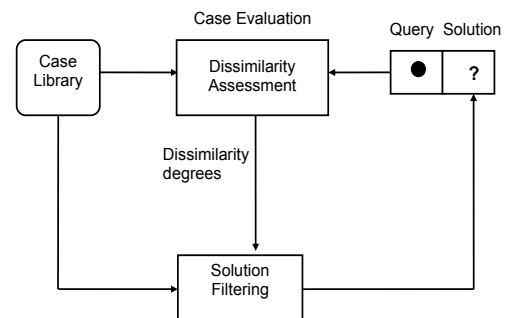


Fig. 1. CBR based on dissimilarity information

In solution filtering, we are concerned with estimating the degrees of impossibility of candidate solutions by using the case information from the case library. We assume solutions of cases to be represented by discrete and mutually exclusive labels in the context of this paper. We define the degrees of impossibility contributed by a single case  $C_i$  (from the case library) by

$$P_i(b) = \begin{cases} Dsim(Q, C_i) & \text{if } Solution(C_i) = b \\ 0 & \text{if } Solution(C_i) \neq b \end{cases} \quad (1)$$

where  $b$  represents a candidate solution, and  $Dsim(Q, C_i)$  denotes the degree of dissimilarity between query problem  $Q$  and case  $C_i$ . It bears mentioning that the impossibility degrees in (1) indeed represent a degree of exclusion, which is supported by the observation of the dissimilar case  $C_i$  having solution  $b$ . On the other hand, we will have  $P_i(b)=0$  if  $C_i$  has a solution different from  $b$ , whereas it merely means that no evidence against

solution  $b$  is derived from case  $C_i$ , rather than the support for  $b$  as the solution to query problem  $Q$ .

Next we consider the overall impossibility degrees in light of the whole case library. For calculating the overall degree of impossibility  $Imposs(b)$  for solution  $b$ , we only need to focus on a subset of cases which have that solution. This is owing to the fact that all other cases in the case library contribute no information for the impossibility of solution  $b$ , as indicated in Eq. (1). The ordered weighted averaging (OWA) operators provide a class of aggregation operators lying between *and* and *or* aggregations. Herein we adopt the S-OWA-OR (OR-like) aggregating operators as the parameterized OWA functions to combine the degrees of impossibility given by the individual cases in the case subset. Let  $S_b = \{ i \mid Solution(C_i) = b \}$  denote the set of indices of the cases having solution  $b$ , the overall impossibility value  $Imposs(b)$  is calculated as

$$Imposs(b) = \underset{i \in S_b}{\tilde{\vee}} P_i(b) = (1 - \alpha) \frac{1}{|S_b|} \sum_{i \in S_b} P_i(b) + \alpha \underset{i \in S_b}{\vee} P_i(b) \quad (2)$$

$$0 \leq \alpha \leq 1$$

Finally we select the solution  $b^*$  that has the lowest overall impossibility value as the final solution for query problem  $Q$ , i.e.,

$$b^* = \underset{\forall b}{\arg \min} [ImPoss(b)] \quad (3)$$

### 4 Fuzzy Dissimilarity Model

This section introduces the structure of fuzzy rules that are used as representation of the dissimilarity model. Suppose there are  $n$  relevant features for problems in the underlying domain. A case  $C_i$  in the case library is described by an  $(n+1)$  tuple:  $C_i = (c_{i1}, c_{i2}, \dots, c_{in}, s_i)$  where  $c_{i1}, c_{i2}, \dots, c_{in}$  denote the feature values in this case and  $s_i$  is the corresponding solution. Likewise we use an  $n$ -tuple  $(y_1, y_2, \dots, y_n)$  to represent a query problem  $Q$  with  $y_j$  referring to the value of the  $j$ th feature in the problem. For comparing case  $C_i$  and query problem  $Q$ , we first need to calculate the values of differences  $x_j = |y_j - c_{ij}|$  on every feature  $j$  between them. Such feature differences are then employed as inputs for condition parts of the fuzzy rules, which collectively decide the dissimilarity value of case  $C_i$  with respect to the query problem.

Assume that the fuzzy sets of feature difference  $x_j$  ( $j=1 \dots n$ ) are represented by  $A(j,1), A(j,2), \dots, A(j, q[j])$  and  $q[j]$  is the number of linguistic terms for  $x_j$ . By  $h(\bullet)$  we denote an integer function

mapping from  $\{1, 2, \dots, m \ (m \leq n)\}$  to  $\{1, 2, \dots, n\}$  satisfying  $\forall i \neq j, h(i) \neq h(j)$ . The fuzzy rules employed in this paper for assessing case dissimilarity are formulated as follows:

$$\text{If } [x_{h(1)} = \bigcup_{k \in D(1)} A(h(1), k)] \text{ and } [x_{h(2)} = \bigcup_{k \in D(2)} A(h(2), k)] \quad (3)$$

$$\text{and} \dots \text{and } [x_{h(m)} = \bigcup_{k \in D(m)} A(h(m), k)]$$

Then Dissimilarity =  $V$

where  $D(i) \subset \{1, 2, \dots, q[h(i)]\}$  for  $i=1 \dots m$ , and  $V \in \{1.0, 0\}$ . Note that the conclusion of the rule in (3) is a singleton being either unity or zero, it can be regarded as a zero-order Sugeno fuzzy rule.

It also bears noting that the premise structure defined above is very general, offering a large degree of flexibility in specification. If the premise of the above rule in (3) includes all input variables in it (e.g.  $m=n$ ), we say that this rule has a complete structure, otherwise its structure is incomplete. Another important feature of the rules in form (3) is that a union of input fuzzy sets is allowed in their premises. Rules having incomplete structure or containing OR connections of input fuzzy sets can achieve larger coverage of input domain, leading to substantial reduction of the number of rules [18, 19].

Finally, with the availability of a set of fuzzy dissimilarity rules in the form of (3), the degree of dissimilarity between case  $C_i$  and query problem  $Q$  can be calculated as follows:

$$Dsim(Q, C_i) = \frac{\sum_{\forall k} t_k(Q, C_i) \cdot V_k}{\sum_{\forall k} t_k(Q, C_i)} \quad (4)$$

where  $V_k$  is the singleton conclusion for rule  $R_k$ , and  $t_k$  denotes the firing strength of rule  $R_k$  when comparing case  $C_i$  and query problem  $Q$ .

### 5 Learning Fuzzy Dissimilarity Rules

We now discuss how to generate fuzzy rules as formulated in the preceding section to build a dissimilarity model. Our aim is to elicit dissimilarity values between cases that can precisely reflect the level of distinction between their solutions. Supervised learning will be performed in this paper to acquire competent fuzzy rules for dissimilarity evaluation. In the following we will first explain how adequate training examples can be created for learning and then we shall outline a genetic algorithm (GA) for automatic generation of fuzzy rules to mimic the training examples.

#### 5.1 Deriving Training Examples

The training examples for fuzzy dissimilarity learning can be created by resorting to the case

library. Since case solutions there are available, it is straight forward to obtain the desired value of dissimilarity for a pair of cases by comparing their solutions. Hence the desired dissimilarity value between case  $C_i$  and  $C_j$  can be defined as:

$$Dsim(C_i, C_j) = Dist(s_i, s_j) \quad (5)$$

where  $Dist$  represents distinction level, and  $s_i$  and  $s_j$  are the solutions of cases  $C_i$  and  $C_j$  respectively.

The criterion for judging distinction level between case solutions is usually domain dependent, thus we cannot further concretize equation (5) without considering problem context and specifics. Nevertheless, in this paper we assume case solutions are represented by discrete labels (e.g. classes), the distinction level between solutions can simply be expressed as follows:

1. If the solutions (labels) have no orders, the distinction level is a binary function as

$$Dist(s_i, s_j) = \begin{cases} 0 & \text{if } s_i = s_j \\ 1 & \text{if } s_i \neq s_j \end{cases} \quad (6)$$

2. If the solutions (labels) have ordinal values, the distinction level should reflect the relative distance in the order. Thus we have

$$Dist(s_i, s_j) = \begin{cases} 0 & \text{if } s_i = s_j \\ \frac{e(s_i, s_j)}{K} & \text{if } s_i \neq s_j \end{cases} \quad (7)$$

where  $K$  is the total number of labels and  $e(s_i, s_j)$  denotes the number of labels between  $s_i$  and  $s_j$  in the order.

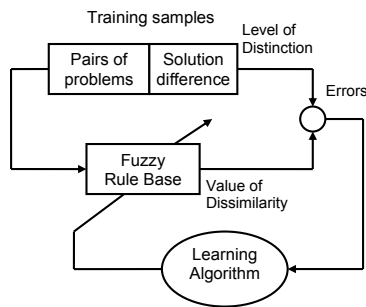


Fig. 2. Fuzzy learning from training samples

The equations (5-7) enable us to acquire many training samples from pairs of cases in the case library. Since we can create a training example for every pair of cases, a much larger multitude of training samples than the number of cases will be created. Next, as shown in Fig. 2, the task of the learning algorithm is to identify optimal fuzzy rules together with associated membership functions such that the dissimilarity degrees assessed via fuzzy

reasoning will comply with the distinction levels specified in the training samples.

### 5.2 Learning Rule Premises by Genetic Algorithms

Learning the fuzzy rules formulated in (3) is solved by identifying suitable premises for different conclusions. For instance, we need to discover under what circumstances two cases in comparison should have a dissimilarity degree of unity. The issue as such is termed as premise learning. In this paper we apply the genetic algorithm (GA) introduced by Godenberg [20] to search for general premises of rules. The purpose is to take advantage of the strength of genetic search to find a set of suitable premise structures together with parameters of associated fuzzy set membership functions. In the following we shall state briefly about coding scheme, genetic operators and fitness function which present key points for the genetic learning of the fuzzy dissimilarity rules.

**Genetic Coding Scheme.** The information concerning structure of rule premises can be considered as a set of discrete parameters, while the information about fuzzy set membership functions is described by a set of continuous parameters. Owing to the different natures between the information about rule structure and about membership functions, a hybrid string consisting of two substrings is used here as the coding scheme. The first substring is a binary code representing premise structure of the fuzzy knowledge base, and the second substring is an integer code corresponding to parameters of fuzzy sets used by the fuzzy rules.

Usually membership functions of a feature difference as input are characterized by a set of parameters. Each of these parameters can further be mapped into an integer through discretization. The resulting integers are then combined to form an integer-vector depicting the fuzzy partition of that input variable. The integer code as one part of the hybrid string is formed by merging together integer-vectors for all inputs (feature differences)

Regarding rule premises, it is easy to see from (3) that premise structure of general rules is decided by integer subsets  $D(i)$  ( $i=1, 2, \dots, m$ ). This fact suggests that a binary code be a suitable scheme for encoding structure of premises, since an integer from  $\{1, 2, \dots, q[h(i)]\}$  is either included in the subset  $D(i)$  or excluded from it. For feature difference  $x_j$  which is included in the premise (i.e.  $h^{-1}(j) \neq \emptyset$ ),  $q[j]$  binary bits need to be used to depict the subset  $D(h^{-1}(j)) \subset \{1, 2, \dots, q[j]\}$ , with bit "1" representing the presence of the corresponding

fuzzy set in the OR-connection and vice versa. If feature difference  $x_j$  does not appear in the premise, i.e.,  $h^{-1}(j)=\emptyset$ , we use  $q[j]$  one-bits to describe the wildcard of “don’t care”. For instance, the condition “if  $[x_1=(small\ or\ large)]$  and  $[x_3=medium]$  and  $[x_4=(medium\ or\ large)]$ ” can be coded by the binary group (101; 111; 010; 011). Further, the whole substring for the premise structure of the rule base is a merge of bit groups for all individual rule premises.

**Crossover.** Owing to the distinct nature between the two substrings, it is preferable that the information in both substrings be mixed and exchanged separately. Here a three-point crossover is used. One breakpoint of this operation is fixed to be the splitting point between both substrings, and the other two breakpoints can be randomly selected within the two substrings respectively. At breakpoints the parent bits are alternatively passed on to the offspring. This means that offspring get bits from one of the parents until a breakpoint is encountered, at which they switch and take bits from the other parent.

**Mutation.** Because of the distinct substrings used, different mutation schemes are needed. Since parameters of input membership functions are essentially continuous, a small mutation with high probability is more meaningful. Therefore it is so designed that each bit in the substring for membership functions undergoes a disturbance. The magnitude of this disturbance is determined by a Gaussian density function. For the binary substring representing the structure of rule premises, mutation is simply to inverse a bit, replace ‘1’ with ‘0’ and vice versa. Every bit in this substring undergoes a mutation with a quite low probability.

**Fitness Function.** An individual (hybrid string),  $HS$ , in the population is evaluated according to its modeling accuracy with respect to the training examples. As many pairs of cases are included in the training data set, we have to consider the total sum of modeling errors for measuring the overall performance of the hybrid string. The total error function is given by

$$Error(HS) = \sum_{(i,j) \in SI} |Dist(s_i, s_j) - Dsim(C_i, C_j)| \quad (8)$$

where  $s_i$  and  $s_j$  are the solutions of cases  $C_i$  and  $C_j$  respectively, and  $SI$  refers to the set of pairs of case indexes corresponding to pairs of cases included in the training data set. At last, the fitness of individual  $HS$  is defined with inverse relation to the mean modeling error as follows

$$Fitness(HS) = 1 - \frac{Error(HS)}{|SI|} \quad (9)$$

## 6 Evaluation Results

We have applied our proposed approach to the problems of classification and diagnosis. In this section we illustrate a case study made on the well-known benchmark problem of wine data classification. The wine data can be downloaded from the address <ftp.ics.uci.edu/pub/machine-learning-databases>. It consists of 178 samples with 13 continuous attributes from three classes.

To test the feasibility of learning fuzzy dissimilarity rules from a small number of cases, we randomly selected 33% of the cases from the Wine data set as the case base used for learning and the remaining cases as test data containing query problems. The fuzzy rules learnt from the case base were then applied for dissimilarity assessment in case-based classification of the problems in the test data set. We performed such tests 10 times.

Table 1 illustrates the classification accuracy on the test data in the 10 tests. It is interesting to observe that, despite the small case bases with about 66 instances in each, very good classification accuracy was still achieved by our CBR system employing the learned fuzzy dissimilarity models.

Table 1: Classification accuracy on test data

Numbers of tests	Classification accuracy
1	97.48%
2	90.76%
3	94.12%
4	93.22%
5	92.44%
6	93.28%
7	91.53%
8	90.76%
9	90.76%
10	94.96%
<b>Average</b>	92.93%

In table 2, we compare our work with some other machine learning approaches in terms of classification accuracy (on test data) and the numbers of cases used for learning. The classification accuracy we obtained is rather close to the best result among the other works. In the other aspect, we employed a much lower number of cases for learning than any other work as indicated in the table. It demonstrates that our system can survive with learning from a small amount of examples. This is an attractive advantage distinguishing our

CBR system from other supervised learning systems for classification.

Table 2: Comparison with other methods

Learning methods	Accuracy	Number of cases for learning
This paper	92.93%	59 ~ 60
C4.5 [21]	90.14%	160 ~ 161
Hu [22]	91.63%	160 ~ 161
MOP-1 [23]	96.01%	160 ~ 161

## 7 Conclusion

The significance of this paper is of two folds. First, we proposed a new CBR approach that exploits dissimilarity information in problem solving. This new approach contributes to avoiding the similarity constraint with traditional CBR systems and thereby facilitating global utilization of more information from the case library. Secondly, we developed a new fuzzy dissimilarity model consisting of fuzzy rules for assessing dissimilarity between cases. We believe that fuzzy rules provide a powerful means to express rich domain knowledge for case evaluation in various situations. Further we have explained and demonstrated how competent fuzzy dissimilarity rules can be acquired from the case library by supervised learning.

### References:

- [1] R. L. D. Mantaras et al., Retrieval, reuse, revision and retention in case-based reasoning, *The Knowledge Engineering Review*, Vol. 20, 2005, pp. 215-240.
- [2] R. Kohavi, P. Langley, and Y. Yun, The utility of feature weighting in nearest neighbor algorithms, *Proc. European Conf. Machine Learning*, 1997.
- [3] D. Wettschereck, and D. Aha, Weighting features, *Proc. 1st Int. Conf. on Case-based Reasoning*, 1995, pp. 347-358.
- [4] A. Bonzano, P. Cunningham, and B. Smith, Using introspective learning to improve retrieval in CBR: A case study in air traffic control, *Proc. 2nd Int. Conf. Case-based Reasoning*, 1997, pp. 291-302.
- [5] N. Cercone, A. An, and C. Chan, Rule-induction and case-based reasoning: Hybrid architectures appear advantageous, *IEEE Trans. Knowledge and Data Engineering*, Vol. 11, 1999, pp. 166-174.
- [6] K. Branting, Acquiring customer preferences from return-set selections, *Proc. 4th Int. Conf. Case-Based Reasoning*, 2001, pp. 59-73.
- [7] L. Coyle, and P. Cunningham, Improving recommendation ranking by learning personal feature weights, *Proc. 7th European Conference on Case-Based Reasoning*, 2004, pp. 560-572.
- [8] J. Jarmulak, S. Craw, and R. Rowe, Genetic algorithms to optimize CBR retrieval, *Proc. European Workshop on Case-Based Reasoning (EWCBR 2000)*, 2000, pp. 136-147.
- [9] H. Ahn, K. Kim, and I. Han, Global optimization of feature weights and the number of neighbors that combine in a case-based reasoning system, *Expert Systems*, Vol. 23, 2006, pp. 290-301.
- [10] N. Xiong, and P. Funk, Building similarity metrics reflecting utility in case-based reasoning, *Intelligent & Fuzzy Systems*, Vol. 17, 2006, pp. 407-416.
- [11] N. Xiong, and P. Funk, Combined feature selection and similarity modeling in case-based reasoning using hierarchical memetic algorithm, *Proc. IEEE World Congress on Computational Intelligence*, 2010, pp. 1537-1542.
- [12] N. Xiong, Towards coherent matching in case-based classification, *Cybernetics and Systems*, Vol. 42, 2011, pp. 198-214.
- [13] R. R. Yager, Case-based reasoning, fuzzy systems modeling and solution composition, *Proc. 2nd Int. Conf. Case-Based Reasoning*, Providence, RI, USA, 1997, pp. 633-643.
- [14] H.-D. Burkhard, and M. M. Richter, On the notion of similarity in case-based reasoning and fuzzy theory, S. K. Pal, T. S. Dillon and D. S. Yeung (Eds.), *Soft Computing in Case-Based Reasoning*, Springer, 2001, pp. 29-45.
- [15] D. Dubois, and H. Prade, Fuzzy set modeling in case-based reasoning, *International Journal of Intelligent Systems*, Vol. 13, 1998, pp. 345-373.
- [16] N. Xiong, A hybrid approach to input selection for complex processes, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 32, No. 4, 2002, pp.532-536.
- [17] N. Xiong, and P. Funk, Construction of fuzzy knowledge bases incorporating feature selection, *Soft Computing*, Vol. 10, No. 9, 2006, pp. 796 – 804.
- [18] N. Xiong, and L. Litz, Learning premises of fuzzy rules for knowledge acquisition in classification problems, *Knowledge and Information Systems*, Vol. 4, 2002, pp. 96-111
- [19] N. Xiong, and L. Litz, Reduction of fuzzy control rules by means of precise learning – method and case study, *Fuzzy Sets and Systems*, Vol. 132, 2002, pp. 217-231.
- [20] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning. New York: Addison-Wesley, 1989.
- [21] J. R. Quinlan, *C4.5: Programs for machine learning*, San Mateo: Morgan Kaufman, 1993.
- [22] Y. C. Hu, Finding useful fuzzy concepts for pattern classification using genetic algorithm, *Information Sciences*, Vol. 175, 2005, 1-19.
- [23] H. Ishibuchi, and Y. Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetic-based machine learning, *International Journal of Approximate Reasoning*, Vol. 44, 2007, pp. 4-31.