Applied Soft Computing 13 (2013) 2057–2064

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy rule-based similarity model enables learning from small case bases

Ning Xiong*

School of Innovation, Design and Engineering, Mälardalen University, SE-72123 Västerås, Sweden

ARTICLE INFO

Article history:
Received 29 January 2010
Received in revised form 6 September 2012
Accepted 18 November 2012
Available online 12 December 2012

Keywords:
Similarity
Case-based reasoning
Fuzzy rules
Learning

ABSTRACT

The concept of similarity plays a fundamental role in case-based reasoning. However, the meaning of "similarity" can vary in situations and is largely domain dependent. This paper proposes a novel similarity model consisting of linguistic fuzzy rules as the knowledge container. We believe that fuzzy rules representation offers a more flexible means to express the knowledge and criteria for similarity assessment than traditional similarity metrics. The learning of fuzzy similarity rules is performed by exploiting the case base, which is utilized as a valuable resource with hidden knowledge for similarity learning. A sample of similarity is created from a pair of known cases in which the vicinity of case solutions reveals the similarity of case problems. We do pair-wise comparisons of cases in the case base to derive adequate training examples for learning fuzzy similarity rules. The empirical studies have demonstrated that the proposed approach is capable of discovering fuzzy similarity knowledge from a rather low number of cases, giving rise to the competence of CBR systems to work on a small case library.

1. Introduction

Case-based reasoning (CBR) presents an important cognitive methodology in Artificial Intelligence, which advocates the use of previous experiences to solve new problems [1]. A fundamental principle that underlies CBR is the hypothesis that similar problems have similar solutions. Hence a CBR system 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.

Similarity assessment plays a key role in CBR in that it decides the quality of retrieved cases. A competent similarity model has to reflect the real utility/relevance of cases for solving new problems [2]. So far a wealth of similarity measures has been established for successful applications of CBR in various real-world scenarios. Cunningham [3] proposed a coherent taxonomy which organized the broad range of similarity mechanisms into the four categories (direct, information-based, transformation-based and emergent measures). The work of this paper belongs to the first category and aims to develop direct similarity models for cases with feature-value representation.

Our objective is to build the similarity model as a knowledge container to guide the CBR process [4]. Fuzzy if–then rules are adopted in this paper as the form of knowledge representation due to the following two reasons. First, fuzzy rules provide a flexible means to express the knowledge and criteria for similarity assessment. Second, fuzzy rule based systems are proved universal approximators [5], able to produce accurate reasoning results for similarity evaluation. The learning of fuzzy similarity rules is implemented by exploiting the case base. We consider the case base a valuable resource with hidden knowledge for similarity learning. A sample of similarity is created from a pair of known cases in which the vicinity of case solutions reflects the similarity of case problems. We do pair-wise comparisons of cases in the case base to derive adequate training examples for learning fuzzy similarity rules. The empirical studies have demonstrated that the proposed approach is capable of discovering fuzzy similarity knowledge from a very limited number of cases, giving rise to the competence of CBR systems to work on a small case library.

The paper is organized as follows. Section 2 discusses related works. Section 3 outlines a general CBR paradigm used in the paper. The fuzzy similarity model for case matching is addressed in Section 4. Then, in Section 5, we discuss the issue of how to learn these fuzzy similarity rules from the case base. In Section 6, we present experimental results for evaluation of the proposed method. Finally, concluding remarks are given in Section 7.

2. Related works

The issue of similarity has received much research attention from the CBR community. Plaza et al. [6] discussed the ways to exploit similarity information for explaining CBR results in classification tasks. They indicated that suitable explanation can be derived from building symbolic descriptions of similar aspects among cases. They also illustrated that symbolic descriptions

* Tel.: +46 21 151716; fax: +46 21 103110.
E-mail address: ning.xiong@mdh.se

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.11.009

of similarity can be utilized to support various steps (including retrieve, reuse, revise and retain) within a CBR process.

Conventionally similarity functions are constructed by means of feature weighting [7]. Different weights are assigned to various features to reflect their importance, and the global similarity evaluation is defined as a weighted sum of the local matching values in individual features. The weights can be adapted in terms of the desired order of retrieved cases given by supervisors, see the works in [8–10]. Some other researchers attempted to search for the best values of weights to optimize the accuracy of the CBR systems [11,12]. However, no matter how weights are derived, the capability of these similarity functions is inherently constrained by their structure as the weighted combination of local matching degrees. This weighted combination makes the restriction that variations in feature weights can only affect case ranking in the similarity skyline which is defined as a subset of cases that are closest to a query problem in the Pareto sense [13], and no case outside the skyline will get preference to be selected for reuse.

A new similarity model without feature weighting was proposed in [14] 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. Further the parameters of such compatibility measures can be learned from the case base in reflection of the true utility of cases. It was also explained and analyzed that this new similarity model can approximate case utility more competently than traditional similarity measures using feature weights.

Similarity has been studied in view of fuzzy theory by some researchers. In [15] the central notion of similarity was treated as a fuzzy relation and fuzzy operations were applied as a tool for building composite similarity measures. The use of generalized aggregation operators such as OWA-operators was suggested by [16] as a more flexible means to combine local similarity values into a global assessment. More recently, the concept of fuzzy similarity was defined and modeled as T-equality in terms of fuzzy residual implications to assess the similarity of fuzzy sets [17].

Dubois et al. [18,19] analyzed the relation between CBR and fuzzy rule-based reasoning and indicated that the fundamental hypothesis of CBR could be formalized in the context of fuzzy rules. It follows that one can consider and implement case-based inference as a special type of fuzzy set-based approximate reasoning. Fuzzy methods were also employed to model user preference and decision principles for case-based recommendation [20]. On the other hand, CBR can be used to assist fuzzy systems as well. In [21] it was demonstrated that CBR could be exploited as feature selection criterion for building compact fuzzy knowledge bases.

3. Case-based reasoning: a general paradigm

The general idea of the case-based approach is exploitation of information in the previous cases to solve a new problem. A general CBR paradigm used in this paper is shown in Fig. 1. It starts with similarity matching between a query problem and known cases in the case library. A properly defined similarity function has to be employed in this stage. As the evaluated similarity values reflect the utility or appropriateness of solutions of the known cases, they offer important information to be utilized in the next step of decision fusion to figure out a final solution for the problem in query.

In decision fusion, we follow the inference rule that “The more similar the two cases are, the more possible it is that their solutions are similar” [20]. Further, we presume a finite number of discrete solutions in the context of this paper. We define the degree of possibility contributed by a single case C_i (i is the index of the case in the case base) by

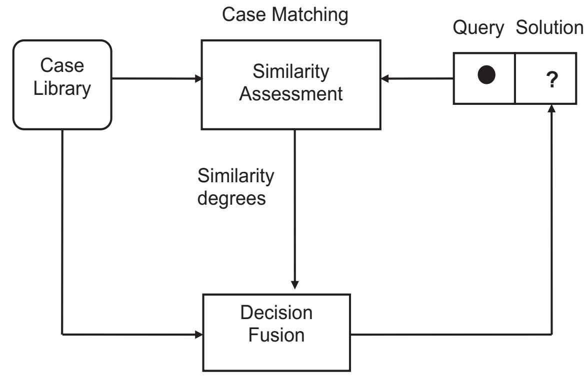

Fig. 1. An overview of the case-based approach.

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

where b represents a candidate solution, and $Sim(Q, C_i)$ denotes the degree of similarity between query problem Q and case C_i. It bears mentioning that the possibility in (1) indeed represents a degree of confirmation, which is supported by the observation that case C_i has a solution identical to 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 information on solution b is derived from case C_i rather than the impossibility of b as the solution to the query problem Q.

Next we consider the overall possibility distribution in terms of the whole case library. For calculating the overall possibility $P(b)$ 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 possibility of solution b, as indicated in Eq. (1). Generally, $P(b)$ should be established as an aggregation of the possibility estimates from the individual cases in this case subset. The ordered weighted averaging (OWA) operators [22] provide a class of aggregation operators lying between “*and*” and “*or*” aggregations. Herein we adopt the S-OWA-OR (OR-like) aggregating operators [23] as the parameterized OWA functions to combine the possibility estimates given by individual cases in the case subset. Let $S_b = \{i | Solution(C_i) = b\}$ denote the set of indices of the cases having solution b, the overall possibility $P(b)$ can be derived via the S-OWA-OR operator ($\tilde{\vee}$) as follows:

$$P(b) = \underset{i \in S_b}{\tilde{\vee}} P_i(b) = (1-\alpha)\frac{1}{|S_b|}\sum_{i \in S_b} P_i(b) + \alpha \max_{i \in S_b}\{P_i(b)\} \quad 0 \leq \alpha \leq 1 \quad (2)$$

where $|S_b|$ is the cardinality of the set S_b.

We calculate the overall possibility value for every candidate solution according to (2). Finally we select the solution b^* that has the largest possibility value among candidates as the solution for query problem Q, i.e.

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

It is clear from Eqs. (1)–(3) that similarity degrees play a central role in possibility assessment and thereby exert crucial influence on the final decisions. Designing competent models for similarity evaluation is paramount to the success of many case-based reasoning systems in practice. In the rest of this paper we will discuss how such a similarity model can be realized by fuzzy rule based reasoning and learning.

4. Fuzzy rules based similarity model

This section explains how similarity matching between cases can be implemented by fuzzy rules based reasoning. We will start with discussing the benefits of fuzzy rules as similarity model in Section 4.1. Then we explain the general rule structure and fuzzy reasoning procedure employed for similarity assessment in Section 4.2.

4.1. The benefits of fuzzy rule based similarity model

We aim to apply fuzzy-rule based reasoning to replace traditional distance based similarity functions in our hybrid CBR system. Every case in the case library is evaluated by fuzzy rules as how much it is similar to a new problem in query. Introducing fuzzy rules as criteria for similarity assessment brings the following significant advantages:

(1) Fuzzy rules appear more flexible to express the knowledge and criteria about similarity between cases than distance-based matching functions. They are therefore capable of making more sophisticated similarity models for complex application domains.
(2) Fuzzy rule-based representation makes it possible for the CBR system to integrate incomplete domain knowledge (if any) in the reasoning procedure. Heuristic expert knowledge can be acquired and formulated into linguistic fuzzy rules by applying knowledge engineering technique. Further the rules acquired from experts can be used as extra rules to supplement the fuzzy rules generated from the case base to achieve more comprehensive knowledge for real-world applications.
(3) Fuzzy linguistic rules are easily understandable for humans. They can well explain how and why a case has been judged as similar to a query problem. Such transparency of the fuzzy similarity model would be important for users to comprehend the procedure and to interact with the CBR system.

4.2. Fuzzy rules and reasoning for similarity assessment

Suppose that there are n features for problems in the underlying domain. A case C_i (i is the case index) in the case base is represented by an $(n+1)$ tuple: $C_i = (y_{i1}, y_{i2}, \ldots, y_{in}, S_i)$ where $y_{i1}, y_{i2}, \ldots, y_{in}$ denote the feature values in this case and S_i is the corresponding solution. In the same manner we use an n-tuple $(z_1, z_2, \ldots, z_n)$ to represent a query problem Q with z_j representing value of the jth feature in the problem. For comparing case C_i and query problem Q, we first need to calculate the values of differences $x_j = |z_j - y_{ij}|$ on every feature j between them. Such feature differences are then employed as inputs for fuzzy-rule based reasoning to decide the similarity value between case C_i and query problem Q.

Assume that the fuzzy sets of feature difference x_j ($j=1\ldots n$) are represented by $A(j, 1), A(j, 2), \ldots, A(j, q[j])$ and $q[j]$ is the number of linguistic terms (fuzzy sets) for x_j. The fuzzy rules employed in this paper for assessing case similarity are formulated as follows:

$$\begin{aligned} &If\ [x_1 = \bigcup_{k \in D(1)} A(1,k)]\ and\ [x_2 = \bigcup_{k \in D(2)} A(2,k)]\ and\ \ldots\ and \\ &\quad [x_n = \bigcup_{k \in D(n)} A(n,k)]\ Then\ Similarity = V \end{aligned} \tag{4}$$

where $D(j) \subseteq \{1, 2, \ldots q[j]\}$ for $j=1\ldots n$, and $V \in \{1.0, 0\}$. As the conclusion of this rule is a singleton being either unity or zero, it can be considered as a zero-order Sugeno fuzzy rule.

The premise structure of the rules in (4) is characterized by the sets $D(j)$ ($j=1\ldots n$). The integers in $D(j)$ correspond to the linguistic terms of x_j that are included with OR-connection in the rule condition. The OR-connection of input fuzzy sets in premises is a nice property for fuzzy rules. Such rules can cover a group of elementary rules that use complete AND connections of single linguistic terms as rule conditions. For instance, the rule "If (x_1 = NZ $\vee$ PZ) and (x_2 = NZ $\vee$ PZ) then Similarity = 1.0" has the premise as illustrated in Fig. 2, and it covers the four elementary rules as follows:

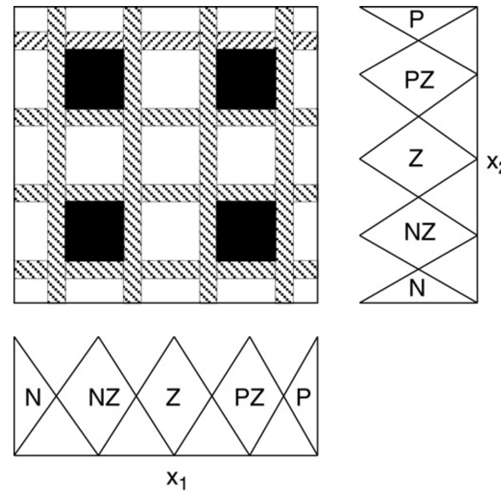

Fig. 2. Rule premise with OR-connection of fuzzy sets.

1) If (x_1 = NZ) and (x_2 = NZ) Then Similarity = 1.0
2) If (x_1 = NZ) and (x_2 = PZ) Then Similarity = 1.0
3) If (x_1 = PZ) and (x_2 = NZ) Then Similarity = 1.0
4) If (x_1 = PZ) and (x_2 = PZ) Then Similarity = 1.0

A special case that may occur with the rules in (4) is the situation when $D(j) = \{1, 2, \ldots q[j]\}$, i.e. all linguistic terms of the input x_j are included in the union for the rule condition. The meaning of this is that we do not care the value of x_j in the premise. It follows that the input x_j is actually excluded from the condition part of the rule for reasoning. If a fuzzy rule includes all input variables in its condition, we say that this rule has a complete structure, otherwise its structure is incomplete.

Rules having incomplete structure or union of fuzzy sets in premises can achieve larger coverage of input domain, leading to substantial reduction of the number of rules [24,25]. Hence they are very beneficial to be employed in complex decision problems with high input dimensionality.

Finally, based on a set of fuzzy similarity rules in the form of (4), we do fuzzy reasoning in the following steps to estimate the degree of similarity between case C_i and query problem Q:

(1) Calculate the vector of feature differences (x_1, x_2, ..., x_n) between Q and C_i, with $x_j = |z_j - y_{ij}|$ for $j=1\ldots n$.
(2) Calculate the firing strength of every rule R_k using the feature differences as inputs

$$t_k(x_1, x_2, \ldots, x_n) = \mu_{PR_k}(x_1, x_2, \ldots, x_n) \tag{5}$$

where $\mu_{PR_k}(x_1, x_2, \ldots, x_n)$ denotes the membership grade of the vector $(x_1, x_2, \ldots, x_n)$ with respect to the premise of rule R_k.
(3) Estimate the degree of similarity by aggregating conclusions of the rules according to their firing strengths. Suppose V_k is the singleton conclusion for rule R_k, the similarity value between query problem Q and case C_i is derived as

$$Sim(Q, C_i) = \frac{\sum_{\forall k} t_k(x_1, x_2, \ldots, x_n) \cdot V_k}{\sum_{\forall k} t_k(x_1, x_2, \ldots, x_n)} \tag{6}$$

5. Learning fuzzy rules from case bases

Supervised learning is performed in this paper to generate fuzzy rules for the similarity model. We need a "teacher" to specify samples of desired similarity values for various pairs of cases as training examples. The task of learning is to revise fuzzy similarity rules to reduce the discrepancy between the desired similarity scores as given by the "teacher" and the estimated similarity values produced by the (similarity) model. The general learning paradigm is shown in Fig. 3 in which a learning algorithm is entailed to adapt the fuzzy similarity model using approximation errors as feedback.

One of the key issues in implementing the above learning paradigm is how to acquire sufficient training examples specified by a "teacher". In many situations it is hard to have a domain expert to do such job due to either economic reasons or the lack of domain knowledge. In the next subsection we shall develop an alternative "teacher" function which exploits the information of cases in the case library to derive training examples for similarity learning.

5.1. Deriving training examples from the case library

Recall that similarity model is used in CBR to identify really useful cases for solving a new problem. We require similarity estimates to reflect the true usefulness of solutions of cases. As shown in Fig. 4, the similarity score $Sim(Q, C_i)$ evaluates the relation between the new problem Q and known case C_i based on problem descriptions while the utility function indicates how useful the solution of case C_i will be for solving the problem Q. What we pursuit is precise approximation of the utility function by the similarity values, i.e.

$$Sim(Q, C_i) = Utility(C_i, Q) \tag{7}$$

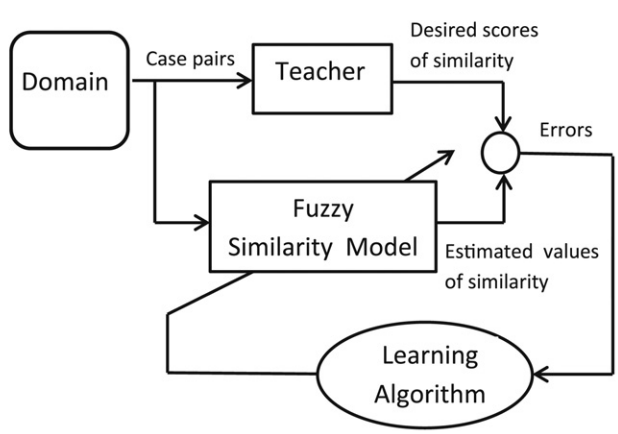

Fig. 3. Supervised learning of the fuzzy similarity model.

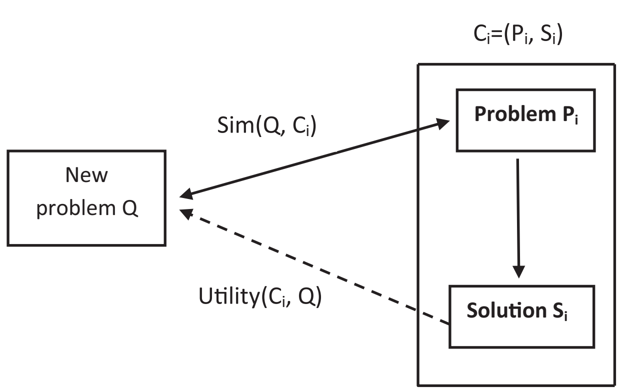

Fig. 4. The similarity and utility measures.

for any query problem Q and case C_i. The relation in (7) implies that the desired similarity value for a pair of cases can be set to equal the degree of utility for one case to solve the problem in the other case.

Further utility values between cases can be derived by comparing case solutions. Let S_i and S_j be the solutions in cases C_i and C_j respectively. The utility of C_i with respect to C_j can be determined by examining the relation between S_i and S_j. The closer solution S_i appears to solution S_j, the more useful solution S_i will be for problem solving in case C_j. In view of this, we define utility between cases as equivalent to the vicinity between their solutions. Thus we can write:

$$Utility(C_i, C_j) = Vic(S_i, S_j) \tag{8}$$

The criterion to assess vicinity between solutions heavily depends on the scenario; hence it is not possible to make more detailed discussion on formula (8) without considering problem context and specifics. Nevertheless, for some applications with discrete and non-ordered solutions, the vicinity between solutions can simply be defined by a binary function as

$$Vic(S_i, S_j) = \begin{cases} 1 & \text{if } S_i = S_j \\ 0 & \text{if } S_i \neq S_j \end{cases} \tag{9}$$

The utility derivation stated above enables us to acquire many utility values for pairs of cases from the case library. They are then used as desired similarity values (for the corresponding case pairs) in the training examples for fuzzy similarity learning. Since we can yield a degree of utility for every pair of cases in the case library, a much larger multitude of training samples than the number of cases can be created. Next, as shown in Fig. 5, the task of the learning algorithm is to evolve the fuzzy knowledge base in the similarity model to produce similarity estimates that comply with the training samples derived from the case library.

5.2. Learning fuzzy rules by genetic algorithms

In this paper we adopt the method of genetic algorithms (GAs) [26] as the learning mechanism to automatically create fuzzy similarity rules from training examples. GAs are stochastic optimization algorithms that emulate the mechanics of natural evolution. They are superior to traditional optimization techniques mainly in the following two aspects. First, a GA evaluates many points in the

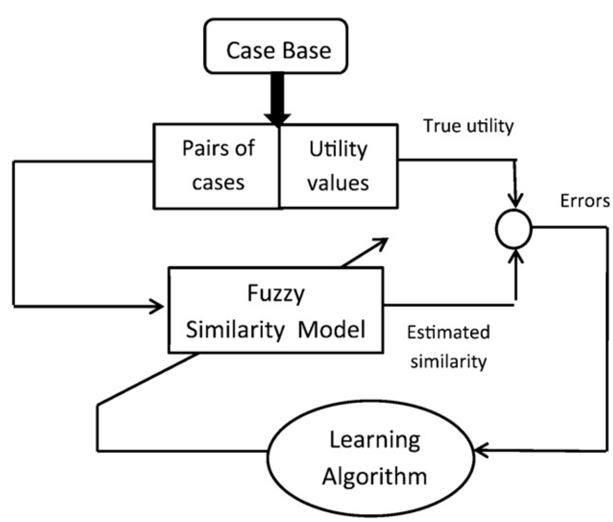

Fig. 5. Fuzzy similarity learning based on training examples derived from the case base.

search space simultaneously, as opposed to a single point, thus reducing the chance of converging to the local optimum. Second, a GA uses only values of objective functions; therefore they do not require the search space to be differentiable or continuous.

Essentially, a GA is an iterative procedure maintaining a constant population size. An individual in the population encodes a possible solution to the problem with a string analogous to a chromosome in nature. At each step of iteration, new strings are created by applying genetic operators on selected parents, and subsequently some of the old weak strings are replaced by new strong ones. In this manner, the performance of the population will be gradually improved in the evolutionary process.

More concretely, the GA is employed here to identify suitable premises of the fuzzy rules in (4) for different conclusions. This problem that the GA handles is termed as premise learning. We want to systematically explore under what circumstances two cases in matching should have a similarity degree of unity and under what situations the value of similarity should be zero. We attempt to utilize the strength of the GA to find best premise structures together with parameters of associated fuzzy membership functions to optimize similarity modeling accuracy in terms of the training examples. In the sequel we shall briefly discuss coding scheme, genetic operators and fitness evaluation which present key points for the genetic learning of rule premises.

Genetic Coding Scheme. The information concerning structure of rule premises can be represented by 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 nature 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 a code with real numbers corresponding to parameters of the fuzzy sets used by the fuzzy rules.

Usually membership functions of a feature difference as input are characterized by a set of continuous parameters. These continuous parameters are combined to form a vector of real numbers depicting the fuzzy partition of that input variable. We then merge together real number vectors for all inputs (feature differences) to get the whole real-valued substring. A code with real-numbers to represent membership functions will deliver higher resolution of solutions than binary or integer codes.

Regarding the code of rule premises, we can see from (4) that the premise structure of a rule is decided by the integer sets $D(j)$ ($j = 1 \ldots n$). Since an integer from $\{1, 2, \ldots, q[j]\}$ is either included in the set $D(j)$ or excluded from it, a binary code appears a natural choice for encoding the structure of premises. For every feature difference x_j, $q[j]$ binary bits are needed to depict the set $D(j) \subseteq \{1, 2, \ldots q[j]\}$ 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, 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. It is worthy to note that the following two situations with a binary group lead to an *invalid premise* encoded: (1) all the bits in the group are equal to one, meaning that no feature differences are considered in the premise and (2) all bits for a feature difference are zero; this input variable thus takes no linguistic term in the premise resulting in an empty fuzzy set in the rule condition part. Rules with **invalid** premises are meaningless, play no role in fuzzy reasoning and therefore have to be discarded. Through elimination of invalid rule premises, the actual rule number can be reduced from the upper limit given by users. This implies an opportunity to adjust the size of the rule base within certain constraints by employing the GA.

Crossover. By the operation of crossover, parent hybrid strings (old fuzzy similarity models) mix and exchange their information through a random process, so that offspring (new fuzzy similarity models) with even higher performance than current individuals will be generated. 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 are 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.

As an illustrative example, consider the two hybrid strings in the following:

$$HS_1 = (b_1^1, b_1^2, b_1^3, b_1^4, b_1^5, b_1^6, b_1^7, b_1^8, b_1^9, b_1^{10}, b_1^{11}, b_1^{12} | c_1^1, c_1^2, c_1^3, c_1^4, c_1^5, c_1^6)$$

$$HS_2 = (b_2^1, b_2^2, b_2^3, b_2^4, b_2^5, b_2^6, b_2^7, b_2^8, b_2^9, b_2^{10}, b_2^{11}, b_2^{12} | c_2^1, c_2^2, c_2^3, c_2^4, c_2^5, c_2^6)$$

Both HS_1 and HS_2 consist of two substrings $(b_i^1, b_i^2, \ldots, b_i^{12})$, $(c_i^1, c_i^2, \ldots, c_i^6)$ $(i = 1, 2)$ representing premise structure and parameters of fuzzy set membership functions respectively. The position between b_i^{12} and c_i^1 is the splitting point between two substrings. Selecting the other two breakpoints for the crossover operator at the position between b_i^5, b_i^6 and the position between c_i^4, c_i^5 respectively, we obtain the two child strings as follows:

$$HS_3 = (b_1^1, b_1^2, b_1^3, b_1^4, b_1^5, b_2^6, b_2^7, b_2^8, b_2^9, b_2^{10}, b_2^{11}, b_2^{12} | c_1^1, c_1^2, c_1^3, c_1^4, c_2^5, c_2^6)$$

$$HS_4 = (b_2^1, b_2^2, b_2^3, b_2^4, b_2^5, b_1^6, b_1^7, b_1^8, b_1^9, b_1^{10}, b_1^{11}, b_1^{12} | c_2^1, c_2^2, c_2^3, c_2^4, c_1^5, c_1^6)$$

Mutation. Mutation is a local operator that transforms the bits of a hybrid string, so as to increase the variability of the population. Because of the distinct substrings used, different mutation schemes are needed. Since parameters of input membership functions are 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 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 accuracy in modeling similarity according 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

$$Error(HS) = \sum_{(i,j) \in SI} |utility(C_i, C_j) - Sim(C_i, C_j)| \tag{10}$$

where SI refers to the set of pairs of case indexes corresponding to pairs of cases included in the set of training examples. In this paper we only focus on problems with discrete solutions, the general error function in (10) can be specialized into

$$Error(HS) = \sum_{(i,j) \in SI} \begin{cases} 1 - Sim(C_i, C_j) & if\ S_i = S_j \\ Sim(C_i, C_j) & if\ S_i \neq S_j \end{cases} \tag{11}$$

where S_i and S_j are solutions in cases C_i and C_j respectively. Further the fitness of individual HS is defined with inverse relation to the mean modeling error as follows

$$Fitness(HS) = 1 - \frac{Error(HS)}{|SI|} \tag{12}$$

At last, we summarize the GA for learning fuzzy similarity model as follows:

Step 1: Generate initial population Pop(0) composed of randomly generated hybrid strings and evaluate their fitness values with (12).
Step 2: Select parents from current population Pop(t) according to probability distribution based on fitness values of individuals.
Step 3: Apply genetic operators on the selected parents to produce a set of offspring.
Step 4: Evaluate the fitness values of the offspring with (12).
Step 5: Choose the L best individuals from the population Pop(t) and the offspring set to form the next generation Pop($t+1$) (L is the population size).
Step 6: Terminate the search procedure if a satisfactory fuzzy similarity model is found or the maximal number of generations is reached. Otherwise go to Step 2.

6. Experimental evaluations

To evaluate the capability of the proposed method, we show in this section the experimental results on six well-known data sets from the UCI Machine Learning Repository [27]. All these data sets contain cases characterized by numerical features and discrete classes, with the numbers of features ranging from 4 to 13 and the numbers of classes between 2 and 6, as illustrated in Table 1. We used the classification accuracy based on small case bases as the criterion to evaluate the learning ability of our proposed approach. We also compared our work with some other machine learning techniques on the same data sets according to both classification accuracy and the number of cases used for learning.

6.1. Scheme of learning in experiments

We constructed fuzzy similarity rules by learning from case bases in the experiments. Each feature difference x_j was assigned with three fuzzy sets $A(j, 1)$, $A(j, 2)$, and $A(j, 3)$ when testing with all the six data sets. The membership functions of these three fuzzy sets, as illustrated in Fig. 6, can be interpreted with linguistic terms such as *small*, *medium*, and *large* respectively. The GA was employed to search for the rule premises under different consequences (similarity = 1.0, similarity = 0) and to optimize the parameters (corresponding to the circle in Fig. 6) of the fuzzy set membership functions at the same time. The objective of the GA was to find the optimal solution (fuzzy knowledge base) to maximize the fitness function in (12). The learnt fuzzy rules were used in fuzzy reasoning to assess the similarity of known cases with respect to a query problem.

Table 1
The six data sets used in the evaluation.

Data set	Case number	Feature number	Number of classes
Iris	150	4	3
Wine	178	13	3
Pirma	768	8	2
New-Thyroid	215	5	3
Breast-W	699	9	2
Cleveland	297	13	2

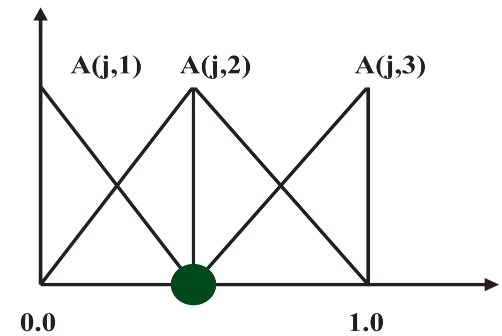

Fig. 6. The three membership functions for feature difference x_j.

Experiments were repeated 20 times on each of the data sets in the evaluation. In every experiment we randomly selected a smaller part of a date set as the case base used for similarity learning and the rest (bigger) part as the test data set providing query problems. The fuzzy rules derived from the case base were used as the similarity model of the CBR system applied to classify problems in the test data.

6.2. Classification results based on small case bases

Since our particular purpose was to examine the learning ability of the proposed approach with a low number of cases, only a small subset (minority) of a data set was selected randomly as the case base and the remaining cases were treated as the test data. The portions of instances entering the case bases are 10% for Pirma and Breast-W data sets, 20% for Cleveland data set, and 33% for Iris, Wine and New-Thyroid data sets. This is in contrast to the common ways of validation in machine learning, where the majority of known examples are usually treated as training data rather than test data. We did the experiments 20 times for all data sets and each time with a randomly selected case base.

Tables 2–7 illustrate the mean accuracy of classification on test data obtained from each of the data sets together with the numbers of cases available for learning. We also give the figures with some other machine learning techniques in these tables for comparison. It is seen from the tables that our method could produce good classification results similar to those reported by others, not to mention that the accuracy on the Cleveland data acquired in this paper is strongly dominating over all others. In the other aspect,

Table 2
The results on the **Iris** data and comparison with others.

The methods	Accuracy on test data	Number of cases used for learning
This paper	0.959	**50**
C4.5 [28]	0.947	135
IGA classifier [29]	0.951	135
Ref. [30]	0.953	135
Ref. [25]	0.967	144

Table 3
The results on the **Wine** data and comparison with others.

The methods	Accuracy on test data	Number of cases used for learning
This paper	0.923	**59–60**
C4.5 [28]	0.901	160–161
IGA classifier [29]	0.937	160–161
Ref. [30]	0.916	160–161
Ref. [31]	0.944	160–161
SOP-3 [32]	0.935	160–161
MOP-3 [32]	0.970	160–161

Table 4
The results on the **Pirma** data and comparison with others.

The methods	Accuracy on test data	Number of cases used for learning
This paper	0.713	**76–77**
C4.5 [28]	0.698	691–692
IGA classifier [29]	0.752	691–692
Ref. [30]	0.693	691–692
Ref. [31]	0.750	691–692
SOP-3 [32]	0.758	691–692
MOP-3 [32]	0.782	691–692

Table 5
The results on the **New-Thyroid** data and comparison with others.

The methods	Accuracy on test data	Number of cases used for learning
This paper	0.907	**71–72**
C4.5 [28]	0.940	193–194
IGA classifier [29]	0.940	193–194
Ref. [30]	0.949	193–194

Table 6
The results on the **Breast-W** data and comparison with others.

The methods	Accuracy on test data	Number of cases used for learning
This paper	0.934	**69–70**
Ref. [25]	0.910	583
C4.5 [28]	0.948	629–630
IGA classifier [29]	0.953	629–630
Ref. [30]	0.949	629–630
Ref. [31]	0.949	629–630
SOP-3 [32]	0.964	629–630
MOP-3 [32]	0.973	629–630

Table 7
The results on the **Cleveland** data and comparison with others.

The methods	Accuracy on test data	Number of cases used for learning
This paper	0.761	**59–60**
Ref. [31]	0.537	267–268
SOP-3 [32]	0.546	267–268
MOP-3 [32]	0.574	267–268

we employed a much lower number of cases for learning than any other work as indicated in the tables. This shows a very interesting merit of our proposed method that it can succeed in learning from a very limited amount of examples. This can be attributed to the pair-wise comparisons of cases in the case library, which produce multiplication of training patterns for fuzzy-rule based similarity modeling.

7. Conclusion

This paper puts forward a new method of employing fuzzy rules as the representation of similarity models in CBR research. Fuzzy rules are considered a more powerful vehicle to accommodate rich domain knowledge than conventional similarity metrics. Fuzzy rule-based reasoning is conducted to estimate the degrees of similarity between cases in the case library and a new problem. Further we explain that fuzzy similarity rules can be generated by exploiting the information from a rather small case library. This is a very attractive advantage and empowers CBR systems to succeed in problem domains where only a low number of experiences are available.

Moreover, we would like to mention that learning of similarity rules from pairs of cases can be considered as a sort of relation-oriented learning. This may be a new contribution to the family of supervised learning. Rather than mimicking the behavior in individual instances as is usually done, the focus of our work is to model the relations between instances that implicitly reside within the set of known cases.

Acknowledgement

The author would like to sincerely thank the anonymous referees for their valuable comments and suggestions for improvement of the paper.

References

[1] R.L.D. Mantaras, et al., Retrieval, reuse, revision and retention in case-based reasoning, The Knowledge Engineering Review 20 (2005) 215–240.
[2] R. Bergmann, M. Richter, S. Schmitt, A. Stahl, I. Vollrath, Utility-oriented matching: a new research direction for case-based reasoning, in: Proc. German Conference on Professional Knowledge Management, 2001, pp. 264–274.
[3] P. Cunningham, A taxonomy of similarity mechanisms for case-based reasoning, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 1532–1543.
[4] M.M. Richter, A. Aamodt, Case-based reasoning foundations, The Knowledge Engineering Review 20 (2006) 203–207.
[5] P. Liu, Mamdani fuzzy system: universal approximator to a class of random processes, IEEE Transactions on Fuzzy Systems 10 (2002) 756–766.
[6] E. Plaza, E. Armengol, S. Ontañón, The explanatory power of symbolic similarity in case-based reasoning, Artificial Intelligence Review 24 (2005) 145–161.
[7] D. Wettschereck, D. Aha, Weighting features, in: Proc. 1st International Conference on Case-Based Reasoning, 1995, pp. 347–358.
[8] K. Branting, Acquiring customer preferences from return-set selections, in: Proc. 4th International Conference on Case-Based Reasoning, 2001, pp. 59–73.
[9] L. Coyle, P. Cunningham, Improving recommendation ranking by learning personal feature weights, in: Proc. 7th European Conference on Case-Based Reasoning, 2004, pp. 560–572.
[10] A. Stahl, T. Gabel, Using evolution programs to learn local similarity measures, in: Proc. 5th International Conference on Case-Based Reasoning, 2003, pp. 537–551.
[11] J. Jarmulak, S. Craw, R. Rowe, Genetic algorithms to optimize CBR retrieval, in: Proc. European Workshop on Case-Based Reasoning (EWCBR 2000), 2000, pp. 136–147.
[12] H. Ahn, K. Kim, I. Han, Global optimization of feature weights and the number of neighbors that combine in a case-based reasoning system, Expert Systems 23 (2006) 290–301.
[13] E. Hullermeier, I. Vladimirskiy, B.P. Suarez, E. Stauch, Supporting case-based retrieval by similarity skylines: basic concepts and extensions, in: Proc. 9th European Conference Case-Based Reasoning, 2008, pp. 240–254.
[14] N. Xiong, P. Funk, Building similarity metrics reflecting utility in case-based reasoning, Journal of Intelligent & Fuzzy Systems 17 (2006) 407–416.
[15] H.-D. Burkhard, M.M. Richter, On the notion of similarity in case-based reasoning and fuzzy theory, in: S.K. Pal, T.S. Dillon, D.S. Yeung (Eds.), Soft Computing in Case-Based Reasoning, Springer, 2001, pp. 29–45.
[16] R.R. Yager, Soft aggregation methods in case based reasoning, Applied Intelligence 21 (2004) 277–288.
[17] H.L. Capitaine, A relevance-based learning model of fuzzy similarity measure, IEEE Transactions on Fuzzy Systems 20 (2012) 57–68.
[18] D. Dubois, H. Prade, Fuzzy set modeling in case-based reasoning, International Journal of Intelligent Systems 13 (1998) 345–373.
[19] D. Dubois, E. Hullermeier, H. Prade, Fuzzy set-based methods in instance-based reasoning, IEEE Transactions on Fuzzy Systems 10 (2002) 322–332.
[20] D. Dubois, E. Hullermeier, H. Prade, Fuzzy methods for case-based recommendation and decision support, Journal of Intelligent Information Systems 27 (2006) 95–115.
[21] N. Xiong, P. Funk, Construction of fuzzy knowledge bases incorporating feature selection, Soft Computing 10 (2006) 796–804.
[22] R.R. Yager, On ordered averaging aggregation operators in multi-criteria decision making, IEEE Transactions on Systems, Man and Cybernetics 18 (1988) 183–190.
[23] R.R. Yager, D.P. Filev, T. Sadeghi, Analysis of flexible structured fuzzy logic controllers, IEEE Transactions on Systems, Man and Cybernetics 24 (1994) 1035–1043.
[24] N. Xiong, L. Litz, Reduction of fuzzy control rules by means of precise learning – method and case study, Fuzzy Sets and Systems 132 (2002) 217–231.
[25] N. Xiong, L. Litz, H. Ressom, Learning premises of fuzzy rules for knowledge acquisition in classification problems, Knowledge and Information Systems 4 (2002) 96–111.
[26] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, New York, 1989.
[27] D. Newman, S. Hettich, C. Blake, C. Merz, UCI Repository of Machine Learning Databases, 1998, URL: http://www.ics.uci.edu/~mlearn/ML.Repository.html
[28] J.R. Quinlan, C.4.5: Programs for Machine Learning, Morgan Kauffman, San Mateo, CA, 1993.

[29] S.Y. Ho, H.M. Chen, S.J. Ho, Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 34 (2004) 1031–1043.
[30] Y.C. Hu, Finding useful fuzzy concepts for pattern classification using genetic algorithm, Information Sciences 175 (2005) 1–19.
[31] T. Elomaa, J. Rousu, General and efficient multisplitting of numerical attributes, Machine Learning 36 (1999) 201–244.
[32] H. Ishibuchi, Y. Nojima, Analysis of interpretability-accuracy trade-off of fuzzy systems by multiobjective fuzzy genetics-based machine learning, International Journal of Approximate Reasoning 44 (2007) 4–31.