A New Case-Based Reasoning Method Based on Dissimilar Relations

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: - Learning relations of objects has recently emerged as a new promising trend for supervised machine learning. Case-based reasoning (CBR) is a subfield of machine learning, which attempts to solve new problems by reusing previous experiences. There is a close link between learning of relations and case-based reasoning in the sense that relation analysis between cases is a core task in a CBR procedure. Traditional CBR systems built upon similar relations can only use local information, and they are restricted by the similarity requirement, i.e., the availability of similar cases to new problems. This paper proposes a new CBR approach that exploits the information about dissimilar relations for solving new problems. A fuzzy dissimilarity model consisting of fuzzy rules has been developed for assessing dissimilarity between cases. Identifying dissimilar cases enables global utilization of more information from the case library and thereby contributes to the avoidance of the similarity constraint with a conventional CBR method. 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 rles; fuzzy reasoning

1 Introduction

Supervised learning presents an important class of machine learning problems [1]. It refers to creating hypotheses or concise knowledge models from training examples to predict the classes or outcomes for future unknown problems. Various learning techniques have been developed to induce knowledge in different representations such as artificial neural networks [2] [3], association rules [4], fuzzy models [5] [6] and support vector machines [7] [8]. So far most works in this area have been devoted to building models to approximate the behavior of single instances residing in the training data set.

Learning relations between objects has recently emerged as a new promising trend for the machine learning research. Rather than focusing on single objects, relation-oriented learning aims to infer predictive models that take into account a pair of objects as input and predict whether the relation of interest exists between both objects [9]. Analysis and modeling of relational data has found wide applications in a number of fields including: computer games [10], bioinformatics [11], social networks [12], financial time series modeling [13], as well information retrieval and text mining [14], to mention a few.

Identifying pairwise relations of objects is also important for solving problems with case-based reasoning (CBR). CBR is a cognitive methodology that aims to solve new problems by reusing previous experiences [15]. A fundamental principle for conventional CBR methods is the heuristic rule 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.

It bears noting that the concept of similarity is closely related to proximity or adjacency. Conventional CBR systems rely on local information residing in similar cases to solve new problems. However there are two drawbacks with such similarity-based methods. First, the retrieved similar neighboring cases sometimes may not contain truly useful solutions for a query problem. Tackling the situations when similar problems don't have similar solutions is a significant challenge for CBR research. Second, the availability of similar cases is not always warranted in reality due to practical limitations, e.g. when the cases in the case base are sparsely distributed or not representative of the entire problem space [16].

This paper investigates a new CBR approach that attempts to identify dissimilar cases for reasoning and problem solving. We consider two cases to be dissimilar as long as they have distinct or "remote" solutions. Analyzing the information about dissimilar relations would foster utilization of more cases in a global perspective, and thereby avoiding the similarity restriction in a traditional CBR system.

The dissimilarity model will be established to identify 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 system 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 evaluation of case relations. Fuzzy rule-based reasoning can be performed to predict whether and to which degree 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 makes an overview of the related works. Section 3 outlines the proposed new CBR approach that reason upon dissimilar information. The fuzzy dissimilarity model to assess case relation is presented in section 4. We then explain, in section 5, 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, concluding remarks are given in section 7

2 Related Works

Identifying similar relations between cases is a key task for conventional CBR systems. So far the main stream of the works for construction of similarity models has been focused on feature weighting [17] [18]. 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 [19]. The probability of ranking principle was utilized in [20] for the assignment of weight values to features. Case-ranking information was utilized in [21] and [22] 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 [23] and [24]. 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 [25] and [26] 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 [27], 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 inconsistence derived from pairwise matching of cases from the case base.

The integration of fuzzy theory with CBR methodology has been addressed by some researchers. Yager [28] explained that there was a close connection between fuzzy system modeling and case based reasoning. Dubois and Prade [29] 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.

Fuzzy set and fuzzy logic have also been used for case representation and case matching in a CBR process. Fuzzy sets were used to depict imprecise case features in a fuzzy case-based reasoning system [30]. In [31] similarity was treated as a fuzzy relation and fuzzy operations were employed for aggregating local similarity values. Moreover, fuzzy linguistic rules were adopted in [32], [33] and [34] as a flexible means to express the criteria for assessing the relation of similarity between cases.

On the other hand, CBR can be used to support fuzzy system modeling as well. In [35] and [36] 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.

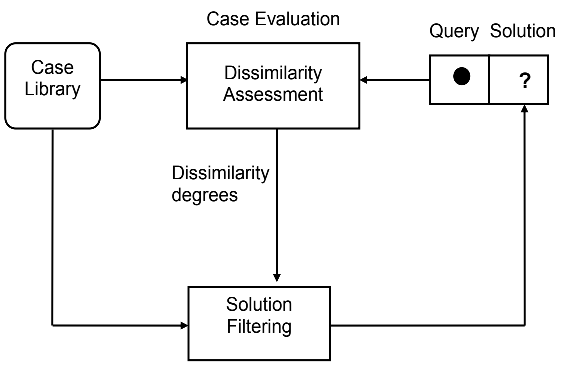

Fig. 1. CBR based on dissimilarity information

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

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

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 $S_b = \{\ i | Solution(C_i) = b\}$ denote the set of indices of the cases having solution b, the overall impossibility value $Imposs(b)$ is calculated as

$$Imposs(b) = \tilde{\vee}_{i \in S_b} P_i(b) = (1-\alpha)\frac{1}{|S_b|}\sum_{i \in S_b} P_i(b) + \alpha \vee_{i \in S_b} P_i(b) \tag{2}$$
$$0 \le \alpha \le 1$$

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

$$b^* = \arg_{\forall b} \min[ImPoss(b)] \tag{3}$$

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: $C_i = (c_{i1}, c_{i2}, \cdots, c_{in}, s_i)$ where $c_{i1}, c_{i2}, \cdots, c_{in}$ denote the feature values in this case and s_i is the corresponding solution. Likewise we use an n-tuple $(y_1, y_2, \cdots, y_n)$ to represent a query problem Q with y_j referring to the 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 $x_j = |y_j - c_{ij}|$ 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… n) are represented by $A(j,1)$, $A(j,2)$, …, $A(j, q[j])$ and $q[j]$ is the number of linguistic terms for x_j. By $h(\bullet)$ we denote an integer function mapping from {1, 2, ..., m ($m \le n$)} to {1, 2, ..., n} satisfying $\forall\ i \ne j$, $h(i) \ne h(j)$. The fuzzy rules employed in this paper for assessing case dissimilarity are formulated as follows:

$$\begin{aligned}&\textit{If}\ \ [x_{h(1)} = \bigcup_{k\in D(1)} A(h(1),k)]\ \ \textit{and}\ \ [x_{h(2)} = \bigcup_{k\in D(2)} A(h(2),k)] \\ &\textit{and} \cdots \textit{and}\ \ [x_{h(m)} = \bigcup_{k\in D(m)} A(h(m),k)] \\ &\textit{Then}\ \ \textit{Dissimilarity} = V\end{aligned} \tag{4}$$

where $D(i) \subset \{1, 2, \ldots, q[h(i)]\}$ for i=1…m, and $V \in \{1.0,\ 0\}$. Note that the conclusion of the rule in (4) 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 (4) 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 (4) is that a union of input fuzzy sets is allowed in their premises. Rules containing such OR connections in conditions cover a group of related rules using complete AND connections of single linguistic terms as rule premises. For instance, the rule ''If (x_1=NZ $\vee$ PZ) and (x_2=NZ $\vee$ PZ) then Dissimilarity=1.0'' has the premise illustrated in Fig.2, and it covers the following four related rules:

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

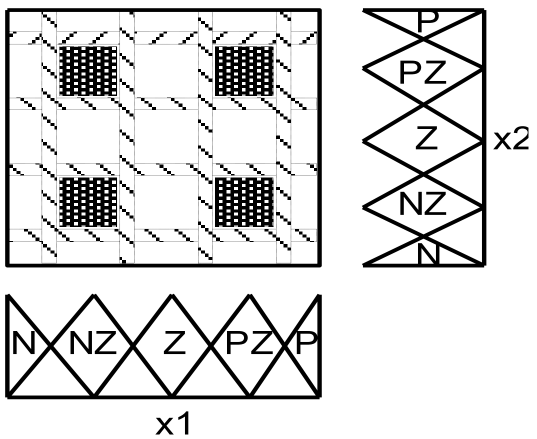

Fig.2. Rule premise with OR-connections

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 [37] [38].

Finally, with the availability of a set of fuzzy dissimilarity rules in the form of (4), the degree of dissimilarity between case C_i and query problem Q can be calculated as follows:

$$Dsim(Q,C_i) = \frac{\sum_{\forall k} t_k(Q,C_i)\cdot V_k}{\sum_{\forall k} t_k(Q,C_i,)} \tag{5}$$

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:

$$Dsim(C_i,C_j) = Dist(s_i,s_j) \tag{6}$$

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 (6) 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

$$Dist(s_i,s_j) = \begin{cases} 0 & \textit{if} \quad s_i = s_j \\ 1 & \textit{if} \quad s_i \ne s_j \end{cases} \tag{7}$$

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

$$Dist(s_i, s_j) = \begin{cases} 0 & if \quad s_i = s_j \\ \dfrac{e(s_i, s_j)}{K} & if \quad s_i \neq s_j \end{cases} \tag{8}$$

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.

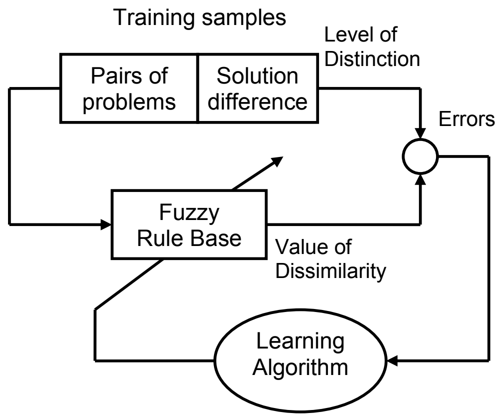

Fig. 3. Fuzzy learning from training samples

The equations (6-8) 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. 3, 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 (4) 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 Goldenberg [39] to search for general premises of dissimilarity 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.

Essentially GAs are global search algorithms that emulate the mechanics of natural genetics and selection. Based on probabilistic decisions they exploit historic information to guide the search for new points in the problem space with expected improvement of performances. In the genetic search a constant population size is always maintained. An individual in the population encodes a possible solution to the problem into a string, which is analogous to a chromosome in nature. At each step of iteration, new strings are created by applying genetic operators on selected parents for recombination. In the sequel we shall briefly discuss the coding scheme, genetic operators and fitness function which are key points for the genetic learning of premises of dissimilarity rules and the associated fuzzy membership functions at the same time.

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 (4) that premise structure of general rules is decided by integer subsets $D(i)$ (i=1, 2, …m). This fact suggests that a binary code be a suitable scheme for encoding structure of premises, since an integer from {1, 2, …, $q[h(i)]$} 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 $D(h^{-1}(j))\subset\{1, 2,...., q[j]\}$, 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.

Example: Consider two strings in the following:

X^1=(0, 0, 1, 1, 1, | 0, 1, 0, 0, 1, 0, 1, | 178, 1678, | 987, 1045)

X^2=(1, 1, 0, 1, 0, | 1, 0, 0, 1, 0, 1, 0, | 982, 1745, | 567, 123)

Both X^1 and X^2 consist of two substrings representing structure of rule premises and parameters of input fuzzy sets respectively. The *12th* position of a hybrid string is the splitting point between two substrings. Selecting the 5*th* and *14th* positions of the hybrid strings to be the other two breakpoints for crossover, we obtain two child strings as follows:

X^3=(0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, | 178, 1678, 567, 123)

X^4=(1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, | 982, 1745, 987, 1045)

Clearly this three-point crossover used here is equivalent to two one-point crossovers operating on both substrings separately.

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

$$Error(HS) = \sum_{(i,j)\in SI} \left| Dist(s_i, s_j) - Dsim(C_i, C_j) \right| \tag{9}$$

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

$$Fitness(HS) = 1 - \frac{Error(HS)}{|SI|} \tag{10}$$

Finally our algorithm for genetic learning of the fuzzy dissimilarity model consists of the following steps:

Step 0 (Initialization): Generate an initial population containing N_{pop} hybrid strings with N_{pop} being the population size.

Step 1 (Initial evaluation): Apply every fuzzy dissimilarity model created initially to the pairs of cases included in the training data set to get assessed dissimilarity degrees. Such dissimilarity values are then compared with the distinction levels specified in the training data to yield an overall fitness score according to (10).

Step 2 (Selection): Select $0.5N_{pop}$ pairs of strings from the current population. The selection probability *prob*(*HS*) for a string *HS* in a population ψ is defined as:

$$\text{prob(HS)} = \frac{\text{Fitness(HS)}}{\sum_{\text{Sr}\in\psi} \text{Fitness(Sr)}} \tag{11}$$

Step 3 (Crossover): Apply the three-point crossover to every pair of selected parents to obtain a set of offspring.

Step 4 (Mutation): Each binary bit of the child strings is reversed with a very low probability, while every integer in the child strings is disturbed with a magnitude determined by a Gaussian density function.

Step 5 (Selective breeding): Employ the fitness function (10) to evaluate each child string in the offspring set. After this, select the best N_{pop} individuals from the current population and the offspring set to form the next generation.

Step 6 (Termination test): If a pre-specified generation number has not been reached, go to Step 2, otherwise terminate the search procedure and return the best individual in the population as the final solution for building the fuzzy dissimilarity model.

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 features from three classes.

Each feature difference x_i (i=1, 2, ..., 13) was assigned with three fuzzy sets $A(i, 1)$, $A(i, 2)$, and $A(i, 3)$ to build fuzzy rules for assessing the dissimilarity between cases. The membership functions of these three fuzzy sets, as illustrated in Fig. 4, can be interpreted with linguistic terms such as *small, medium, and large* respectively. The GA was set into force to search for the rule premises under different consequences (dissimilarity=1.0, dissimilarity=0) and to optimize the parameters (corresponding to the circle in Fig. 4) of the fuzzy set membership functions at the same time. The objective of the GA was to discover the fuzzy knowledge base to maximize the fitness function in (10). The learnt fuzzy rules were employed as evaluation criteria to determine the dissimilarities of cases, which were then used in solution filtering for estimating impossibility degrees of candidate solutions.

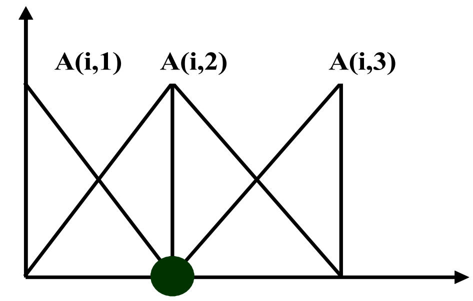

Fig. 4. The three membership functions for the difference on feature x_i

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 used 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%
Average	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 [40]	90.14%	160 ~ 161
Ho [41]	93.72%	160 ~ 161
Hu [42]	91.63%	160 ~ 161
Elomaa [43]	94.40%	160 ~ 161
SOP-1 [44]	92.70%	160 ~ 161
MOP-1 [44]	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 have explained and demonstrated how competent fuzzy dissimilarity rules can be acquired from the case library by supervised learning. Introducing fuzzy rules to evaluate pairwise relations of cases would bring the following advantages:

- Fuzzy rules provide a powerful and flexible means to express the criteria for evaluation of case relations in various situations. Owing to the rich information contained in the fuzzy dissimilarity model, large amount of dissimilar cases will be recognized for usage in the step of solution filtering
- The knowledge base containing fuzzy rules makes it possible to acquire and integrate knowledge from multiple sources. The fuzzy dissimilarity rules generated from the case library could be supplemented by extra rules defined by domain experts to achieve more comprehensive knowledge in real applications.
- The fuzzy knowledge base is well comprehensible by means of interpretation of individual linguistic rules. It follows that human users can easily understand how cases have been assessed as dissimilar by checking involved fuzzy rules. This creates opportunity for human involvement and interaction with the CBR system to dynamically adapt system behaviour to any specific demands.

Acknowledgements

The work is supported by the EMOPAC project (project no 16317) and the ESS-H profile, both funded by the Swedish Knowledge Foundation.

References:

[1] S. B. Kotsiantis, Supervised machine learning: A review of classification techniques, *Informatica*, Vol. 31, 2007, pp. 249-268

[2] H. Sug, The effect of training set size for the performance of neural networks of classification, *WSEAS Transactions on Computers*, Vol. 9, 2010, pp. 1297-1306.

[3] Z. Dong, Intelligence diagnosis method based on particle swarm optimized neural network for roller bearings, *WSEAS Transactions on Systems*, Vol. 12, 2013, pp. 667-677.

[4] J. Hu, and X.-Y. Li, Association rules mining including weak-support modes using novel measures, *WSEAS Transactions on Computers*, Vol. 8, 2009, pp. 559-568.

[5] N. Xiong, Evolutionary learning of rule premises for fuzzy modeling, *International Journal of Systems Science*, Vol. 32, 2001, pp. 1109-1118.

[6] S. Vassileva, Advanced fuzzy modeling of integrated bio-systems, *WSEAS Transactions on Systems*, Vol. 11, 2012, pp. 234-243.

[7] L. State, and I. Paraschiv-Munteanu, SVM-based supervised and unsupervised classification schemes, *WSEAS Transactions on Computers*, Vol. 9, 2010, pp. 1212-1223.

[8] P. Hajek, Forecasting stock market trend using prototype generation classifiers, *WSEAS Transactions on Systems*, Vol. 11, 2012, pp. 671-680.

[9] W. Waegeman, T. Pahikkala, A. Airola, T. Salakoski, M. Stock, and B. D. Baets, A Kernel-based framework for learning graded relations from data, *IEEE Transactions on Fuzzy Systems*, Vol. 20, 2012, pp. 1090-1101.

[10] M. Bowling, J. Furnkranz, T. Graepel, and R. Musick, Machine learning and games, *Machine Learning*, Vol. 63, 2006, pp. 211-215.

[11] Y. Yamanishi, J. Vert, and M. Kanehisa, Protein network inference from multiple genomic data: A supervised approach, *Bioinformat.* Vol. 20, 2004, pp. 1363-1370.

[12] B. Taskar, M. Wong, P. Abbeel, and D. Koller, Link prediction in relational data, *Proc. Adv. Neural Inf. Process. Syst. Conf.* 2004, pp. 659-666.

[13] F. Neri, Quantitative estimation of market sentiment: a discussion of two alternatives, *WSEAS Transactions on Systems*, Vol. 11, 2012, pp. 691-702.

[14] Y. Yang, N. Bansal, W. Dakka, P. Ipeirotis, N. Koudas, and D. Papadias, Query by document, *Proc. 2nd ACM Int. Conf. Web Search data mining,* 2009, pp. 34-43.

[15] 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.

[16] M. W. Floyd, and B. Esfandiari, An active approach to automatic case generation, *Proc. International Conference on Case-Based Reasoning*, 2009, pp. 150-164.

[17] R. Kohavi, P. Langley, and Y. Yun, The utility of feature weighting in nearest neighbor algorithms, *Proc. European Conf. Machine Learning,* 1997.

[18] D. Wettschereck, and D. Aha, Weighting features, *Proc. 1st Int. Conf. on Case-based Reasoning*, 1995, pp. 347-358.

[19] 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.

[20] 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.

[21] K. Branting, Acquiring customer preferences from return-set selections, *Proc. 4th Int. Conf. Case-Based Reasoning*, 2001, pp. 59-73.

[22] 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.

[23] 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.

[24] 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.

[25] N. Xiong, and P. Funk, Building similarity metrics reflecting utility in case-based reasoning, *Intelligent & Fuzzy Sys*tems, Vol. 17, 2006, pp. 407-416.

[26] 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.

[27] N. Xiong, Towards coherent matching in case-based classification, *Cybernetics and Systems*, Vol. 42, 2011, pp. 198-214.

[28] 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.

[29] D. Dubois, and H. Prade, Fuzzy set modeling in case-based reasoning, *International Journal of Intelligent Systems*, Vol. 13, 1998, pp. 345-373.

[30] V. Marques, J. T. Farinha, and A. Brito, Case-based reasoning and fuzzy logic in fault diagnosis, *WSEAS Transactions on Computers*, Vol. 8, 2009, pp. 1408-1417.

[31] 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.

[32] N. Xiong, Fuzzy rule-based similarity model enables learning from small case bases, *Applied Soft Computing*, Vol. 13, 2013, pp. 2057-2064.

[33] N. Xiong, Learning fuzzy rules for similarity assessment in case-based reasoning, *Expert Systems and Applications*, Vol. 38, 2011, pp. 10780-10786.

[34] N. Xiong, L. Ma, and S. Zhang, Fuzzy relational learning: a new approach to case-based reasoning, *Proc. 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2013),* Shengyang, China, 2013, pp. 612-617.

[35] 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.

[36] N. Xiong, and P. Funk, Construction of fuzzy knowledge bases incorporating feature selection, *Soft Computing*, Vol. 10, No. 9, 2006, pp. 796 – 804.

[37] 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

[38] 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.

[39] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley, 1989.

[40] J. R. Quinlan, *C4.5: Programs for machine learning*, San Mateo: Morgan Kauffman, 1993.

[41] S. Y. Ho, H. M. Chen, and S. J. Ho, Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space, *IEEE Trans. Syst., Manv Cybern., Part B*, Vol. 34, 2004, pp. 1031-1043.

[42] Y. C. Hu, Finding useful fuzzy concepts for pattern classification using genetic algorithm, *Information Sciences*, Vol. 175, 2005, 1-19.

[43] T. Elomaa, and J. Rousu, General and efficient multisplitting of numerical attributes, *Machine Learning*, Vol. 36, 1999, pp. 201-244.

[44] 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.