2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Data-based Fuzzy Rules Extraction Method for Classification

Xinyu Qiao
the Research Center of
Information and Control
Dalian University of Technology
Dalian, China 116024
Email: sherry880228@gmail.com

Zhenying Li
the Research Center of
Information and Control
Dalian University of Technology
Dalian, China 116024
Email: lizhenying_ok@mail.dlut.edu.cn

Wei Lu
and Xiaodong Liu
the Research Center of
Information and Control
Dalian University of Technology
Dalian, China 116024
Email: xdliuros@hotmail.com
Telephone:86-0411-84709380

***Abstract*—In this study, a two-stage method which extracts fuzzy rules directly from samples is proposed for classification. First, we introduce a neighborhood based attribute significance algorithm to select *r* of the most important attributes from the original attribute set. Second, the proposed algorithm generates fuzzy rule from each sample described by the selected attribute subset and finally simplifies the returned fuzzy rule-base. A confidence degree is assigned for each of the extracted fuzzy rules by counting the number of training samples covered by the rule to solve the conflicts among the rules and then the rule-base is pruned. The performance of the proposed classification method have been compared with other five classification approaches including C4.5, DTable, OneR, NNge, and PART on seven UCI data sets. The experimental results show that the proposed method is better than other methods in two aspects: the higher classification accuracy and the smaller rule-base.**

I. Introduction

Many methods have been proposed to manage real-world problems which deal with classification tasks [1]. Fuzzy rule based classification systems (FRBCSs) [2], for their interpretability, is considered an efficient method among the computational intelligence techniques. They have been used in many practical classification problems such as medical applications [3], classification of battlefield ground vehicles [4] or intrusion detection [5]-[6], *etc*. The most challenging problem in the design of FRBCSs is the construction of rule-base for a specific problem. There are many methods proposed to construct the rule-base from numerical data, such as heuristic approaches [7]-[8], neuro-fuzzy techniques [9]-[11], clustering methods [12]-[13], genetic algorithms [14]-[18], and data mining techniques [19]-[21], *etc*. In our study, we propose a data-based fuzzy rules extraction method (DBFREM) which extracts fuzzy rules directly from the input samples. To get a concise rule-base, a new attribute selection algorithm named NBASA (neighborhood based attribute significance algorithm) is proposed to select the most r important attributes from original attribute set. After attribute selection, we introduce a four-step procedure for generating fuzzy rules from the training samples and show how to use these fuzzy rules to obtain a mapping from attribute space to class label set. *Step* 1 divides each attribute into fuzzy regions; *Step* 2 forms a fuzzy rule-base from the training samples; *Step* 3 simplifies the obtained rule-base by using a confidence degree and a pruning algorithm; *Step* 4 presents a reasoning procedure for obtain a mapping based on the concise fuzzy rule-base. The proposed technique combines the advantages of attribute subset selection and data-based classification. To offer a thorough comparative analysis, we experiment with the proposed algorithm using a number of well known data sets coming from the UCI Repository of Machine Learning data [22]. The comparative analysis involves some other types of classifiers such as C4.5 decision tree [23], Decision Table (DTable) [24], One Rule (OneR) [25], Nearest neighbor like Algorithm (NNge) [26], and PART Decision List (PART) [27]. The main attributes of the proposed DBFREM classifier which distinct it from the other methods can be highlighted as follows:

- For most of the datasets, the relevance of classification results perform better than the methods mentioned above such as C4.5, DTable, and PART on accuracies and the size of rule-base.
- The DBFREM algorithm, requiring no mathematical model constructs directly from "samples" and the rule-base extracted by the system is much simpler, easier to understand and easier to interpretable.

The rest of this paper is organized as follows. In Section II, the details of the NBASA algorithm and the DBFREM algorithm are introduced. In Section III, the simulation experiments are presented and the results have been compared with other methods. Section IV concludes the paper.

II. Proposed Algorithm

In this section, we will construct two algorithms for feature selection and rule-base extraction respectively, which compose the data based classification system. Giving a set of samples U, described with condition attribute set C and class label set D (decision), the task of our method is to construct a mapping

This work was supported by the National Natural Science Foundation of China under grants 61175041 and 61034003.

from the condition attribute set to the class label set based on the set of training samples.

A. Attribute Selection

To get a simple and effective rule-base, feature selection is a helpful step in model construction. Especially for the samples described by large number of attributes, it is an effective technique to reduce classification cost and improve generalization in the construction of the classifier.

1) Neighborhood Based Rough Set Model: In this part we mainly review the basic concepts and theoretical results of neighborhood based rough set model. More details can be found in [28]. Formally, the data can be written as a $IS = \langle U, C, D \rangle$, where U is the nonempty set of samples $\{x_1, x_2, x_3, ..., x_n\}$, called a sample space, C is the set of attributes $\{c_1, c_2, c_3, ..., c_m\}$, D is the class label set. Each sample in the sample space is assigned with a subset of samples. The samples in the neighborhood are near the center sample measured with some distance function. The subset is called a neighborhood information granule. The family of neighborhood granules forms a cover of the sample space.

***Definition** 1:* Given arbitrary $x_i \in U$ and $B \subseteq C$, the neighborhood $\delta_B(x_i)$ of x_i in the subset B is defined as

$$\delta_B(x_i) = \{x_j \mid x_j \in U, \Delta_B(x_i, x_j) \leq \delta\} \tag{1}$$

where Δ is a weighted distance function.

The size of the neighborhood depends on the threshold δ. The greater δ is, the more samples will fall into the neighborhood. Neighborhood relations draw the samples together for similarity or indistinguishability in terms of distances.

The neighborhood conception divides the samples into two groups: *positive region* and *boundary*. *Boundary* is the sample subset whose neighborhoods come from more than one decision class. On the other hand, *positive region*, denoted by $POS_B(D)$, is the subset of samples whose neighborhoods consistently belong to one of the classes.

Fig.1 shows an example of binary classification in two condition attribute subset $\{c_1, c_2\}$, where $d \in D$, d_1 is labeled with "plus" and d_2 is labeled with "point". Consider samples x_1, x_2, x_3, we assign circle neighborhoods to these samples. We can find $\delta_B(x_1) \subseteq d_1$ and $\delta_B(x_3) \subseteq d_2$, while $\delta_B(x_2) \cap d_1 \neq \varnothing$, $\delta_B(x_2) \cap d_2 \neq \varnothing$. According to the above definitions: $x_1 \in$ *positive regions*, $x_3 \in$ *positive regions* and $x_2 \in$ *boundary*. The samples in different attribute subsets will have different *boundaries*. The size of the *boundary* reflects the discriminability of the classification problem in the corresponding subsets. It also reflects the recognition power or characterizing power of the condition attributes. The greater the *boundary* region is, the weaker the characterizing power of the condition attributes will be. It can be formulated as follows:

***Definition** 2:* The dependency degree of D to B is defined as the ratio of consistent samples:

$$\gamma_B(D) = |POS_B(D)|/|U| \tag{2}$$

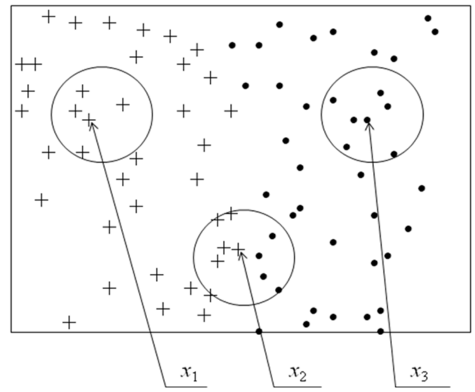


Fig. 1. An example with two classes

where $\gamma_B(D)$ reflects the ability of B approximate to D. Obviously, $0 < \gamma_B(D) < 1$, we say that D completely depends on B if $\gamma_B(D) = 1$.

***Definition** 3:* Given a decision system $\langle U, C, D \rangle$, $B \subseteq C$, $a \notin B$, we define the significance of an attribute as

$$SIG(a, B, D) = \gamma_{B \cup a}(D) - \gamma_B(D) \tag{3}$$

The attribute's significance is the function of three variables: a, B, and D. An attribute a may be of great significance in B_1 but of little significance in B_2.

2) Feature Selection Based on Neighborhood: In this part, we search an attribute subset based on the neighborhood rough set model through the proposed algorithm. For the attribute set $\{c_1, c_2, c_3, ..., c_m\}$, there are 2^m combinations of attribute subsets. It is not practical to search all of the reducts in 2^m combinations. Fortunately, in practice, we usually just require one of the reducts to train a classifier, and we do not care much whether the reduct is really the minimal one or not. Then a tradeoff solution can be constructed, such as greedy forward search algorithm. The NBASA algorithm is constructed as TABLE I.

Here the NBASA algorithm adds an attribute with the great significance into the reduction in each circle until the number of attributes in the reduction meets requirements.

B. Generating Fuzzy Rules From Numerical Data

Suppose we are given a set of training samples. The simple three-attribute case is chosen in order to emphasize and to clarify the basic idea of our new approach.

$$(c_1^{(1)}, c_2^{(1)}, c_3^{(1)}; d^{(1)}), (c_1^{(2)}, c_2^{(2)}, c_3^{(2)}; d^{(2)}), ... \tag{4}$$

where c_1, c_2, c_3 are the condition attributes and d is the class label of one given sample. The task here is to generate a set of fuzzy rules directly from the given samples of (4), and utilize these fuzzy rules to determine a mapping: $f: \{c_1, c_2, c_3\} \Rightarrow D$.

Our approach consists of the following four steps:

Step 1−Divide the Attribute Spaces into Fuzzy Regions

Assume that the intervals of the attributes c_1, c_2, and c_3 are $[c_1^{\min}, c_1^{\max}]$, $[c_2^{\min}, c_2^{\max}]$, and $[c_3^{\min}, c_3^{\max}]$ respectively.

TABLE I
ALGORITHM OF FEATURE SELECTION

Algorithm: NBASA
Input: *Sample*, δ
Sample: the dataset described with condition attribute set C and a class label set of D;
δ: the threshold to control the size of neighborhood;
Output: *Red*
Red: the attribute subset with r attributes.
1. $Red = \mathbf{NBASA}(Sample, \delta)$
2. $red = \varnothing$;
3. **for** each attribute $c_i \in C - red$
4. compute $SIG(c_i, red, D) = \gamma_{red \cup c_i}(D) - \gamma_{red}(D)$; // Here we define $\gamma_{\varnothing}(D) = 0$;
5. **end**
6. $SIG(c_k, red, D) = \max_i\{SIG(c_i, red, D)\}$;
7. $red = red \cup c_k$;
8. **if** the number of attributes in *red* is less than r
9. go to 3;
10. **else**
11. *Red* = *red*;
12. **end**

There are various shapes of membership functions such as Gaussian, trapezoid and all kinds of divisions of the regions. For simplicity, we divide each interval into 3 regions, denoted by S (Small), CE (Center), B (Big), and assign each region a triangular membership function. Fig. 2 (a), (b), (c) show an example where the intervals of c_1, c_2, and c_3 are divided: one vertex lies at the center of the region and has membership value unity; the other two vertices lie at the centers of the two neighboring regions, respectively, and have membership values equal to zero.

Step 2−*Generate a Fuzzy Rule-base from Sample Space*

First, determine the degrees of the attributes $c_1^{(i)}$, $c_2^{(i)}$, and $c_3^{(i)}$ in different fuzzy sets. For example, $c_1^{(1)}$ in Fig. 2 (a) has degree 0.8 in S, degree 0.2 in CE, and zero degree in B. Similarly, $c_2^{(1)}$ in Fig. 2 (b) has degree 1 in CE, and zero degrees in the other two fuzzy sets.

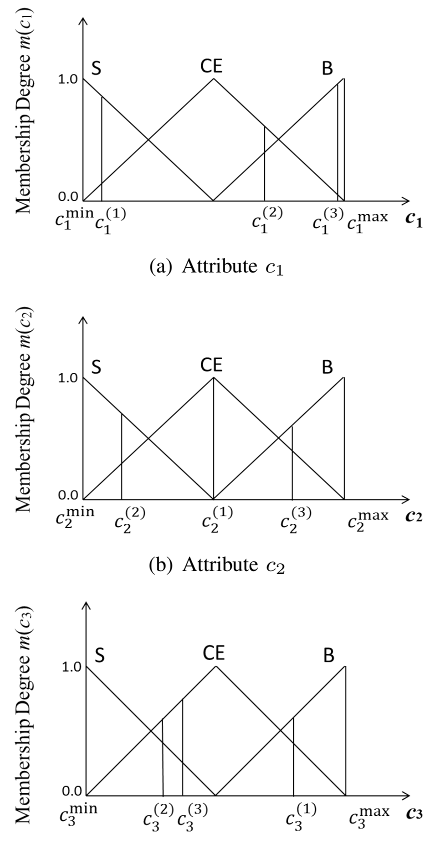
Second, assign a given $c_1^{(i)}$, $c_2^{(i)}$, or $c_3^{(i)}$ to the fuzzy set with maximum degree. For example, $c_1^{(1)}$ in Fig. 2 is considered to be S, and $c_2^{(1)}$ in Fig. 2 is considered to be CE.

Finally, extract one rule from one given sample, suppose $d^{(1)}$ is *Class* 1 and $d^{(2)}$ is *Class* 3, e.g.,

$$(c_1^{(1)}, c_2^{(1)}, c_3^{(1)}; d^{(1)}) \Rightarrow [c_1^{(1)}(0.8 \text{ in } S, \max), c_2^{(1)}(1 \text{ in } CE, \max), c_3^{(1)}(0.6 \text{ in } B, \max);\ d^{(1)} \text{ is } Class\ 1] \Rightarrow Rule\ 1:$$

IF c_1 is S, c_2 is CE, and c_3 is B, THEN the sample is *Class* 1 ([*S*, *CE*, *B*; *Class* 1]);

$$(c_1^{(2)}, c_2^{(2)}, c_3^{(2)}; d^{(2)}) \Rightarrow [c_1^{(2)}(0.6 \text{ in } CE, \max), c_2^{(2)}(0.7 \text{ in } S, \max), c_3^{(2)}(0.6 \text{ in } CE, \max);\ d^{(2)} \text{ is } Class\ 3] \Rightarrow Rule\ 2:$$



(a) Attribute c_1

(b) Attribute c_2

(c) Attribute c_3

Fig. 2. Divisions of the attribute spaces into fuzzy regions and the corresponding membership functions

IF c_1 is CE, c_2 is S, and c_3 is CE, THEN the sample is *Class* 3 ([*CE*, *S*, *CE*; *Class* 3]);

In this way we get a rule-base which contains all of the rules extracted from original samples.

Step 3−*Simplify the Rule-base*

There are two processes in simplifying the rule-base. First, a confidence degree is assigned for each of the extracted fuzzy rules. Then the rule-base is pruned.

Since there are usually lots of samples, and each sample generates one rule, it is highly probable that there will be some conflicting rules, i.e., rules that have the same antecedent part but a different decision labels. For example, rules [*CE*, *S*, *CE*; *Class* 3] and [*CE*, *S*, *CE*; *Class* 1] are two conflicting rules. One way to resolve this conflict is to assign a confidence degree to each rule generated from samples,

First we define a confidence degree to each rule extracted in *step* 2 by counting the number of training samples which the identical rule extracted from. Suppose the total number of samples in the sample space is p, and there are q samples covered by *Rule i*, we define the confidence degree of *Rule i* as

$$D(Rule\ i) = q/p \tag{5}$$

Each rule is assigned with a degree via (5), and we accept only

TABLE II
ALGORITHM OF PRUNING

Algorithm: Pruning the rule-base
Input: *Sample*, *Rule-base*
Sample: the training dataset with l classes;
Rule-base: the rule-base with t rules which was simplified in the first process;
Output: *Final Rule-base*
Final Rule-base: the final rule-base;
1. *Final Rule-base* $=$ **Pruning**(*Sample*, *Rule-base*)
2. $A = \varnothing$;
3. acc $=$ the accuracy of the present rule-base on *Sample*;
4. **for** i= 1 : t
5. delete the i-th rule in the *Rule-base*;
6. a_i $=$ the accuracy of the new rule-base on *Sample*;
7. $A = A \cup a_i$;
8. **end**
9. $a_k = \max\{A\}$;
10. **if** $a_k \geq acc$
11. delete the k-th rule in *Rule-base*;
12. go to 1;
13. **end**
14. *Final Rule-base* $=$ *Rule-base*;
15. **return**

the rule from a conflict group that has the maximum degree. In this way, not only the conflict problem is resolved, but also the number of rules is greatly reduced.

Second the rule-base is pruned to cut down the redundant rules. The rules extracted from the training samples, however, may include redundant structures as well as poorly performing rules, which should be removed from the rule-base to enhance an overall performance of the classifier and improve its efficiency. In what follows, we prune the rule-base by making use of the available training samples:

1) Remove each rule from the rule-base, and classify the training samples using the remaining rules.
2) Delete the rule, whose corresponding remaining rules have the maximal increase of accuracy on training samples.
3) Repeat 1-2 and terminate the pruning if the resulting pruned rule-base becomes worse than the original one when applied to the training samples.

TABLE II presents the pruning algorithm of the rule-base.

Step 4$-$*Determine a Mapping Based on the Fuzzy Rule-base*

We use the following strategy to determine the class label D for given attribute space $\{c_1, c_2, c_3\}$: For each attribute in the given attribute $(c_1^{(i)}, c_2^{(i)}, c_3^{(i)})$, we first apply the strategy mentioned in *step* 2 to decide the fuzzy set of each attribute, for example, in Fig. 2,

$$[c_1^{(3)}(0.9 \text{ in } B, \max), c_2^{(3)}(0.6 \text{ in } B, \max), c_3^{(3)}(0.7 \text{ in } CE, \max)] \Rightarrow (B, B, CE).$$

Then we search the rule-base produced in *step* 3 to find a fuzzy rule whose antecedent part is the same with (B, B, CE) to determine the class label $d^{(i)}$.

In all the antecedents parts of the rule-base, there is a huge possibility that there might not exist a (B, B, CE), in this case, we only consider two attributes from the attribute $(c_1^{(i)}, c_2^{(i)}, c_3^{(i)})$. There are three combination forms, these are $(c_1^{(i)}, c_2^{(i)}, **)$, $(c_1^{(i)}, **, c_3^{(i)})$, and $(**, c_2^{(i)}, c_3^{(i)})$, corresponding combination regions are (B, B, **), (B, **, CE), and (**, B, CE). Then we combine the attributes in each combination using the following product strategy to determine a degree m, i.e.,

$$m(c_1^{(i)}, c_2^{(i)}, c_3^{(i)}) = \sum m(c_j^{(i)}) \tag{6}$$

where $m(c_j^{(i)})$ denotes the maximal degree of attribute j, e.g., the situations $(c_1^{(i)}, c_2^{(i)}, **)$, $(c_1^{(i)}, **, c_3^{(i)})$, and $(**, c_2^{(i)}, c_3^{(i)})$ give

$$m(c_1^{(i)}, c_2^{(i)}, c_3^{(i)}) = m(c_1^{(i)})m(c_2^{(i)}) = 0.9\times0.6=0.54;$$
$$m(c_1^{(i)}, c_2^{(i)}, c_3^{(i)}) = m(c_1^{(i)})m(c_3^{(i)}) = 0.9\times0.7=0.63;$$
$$m(c_1^{(i)}, c_2^{(i)}, c_3^{(i)}) = m(c_2^{(i)})m(c_3^{(i)}) = 0.6\times0.7=0.42.$$

For each rule whose antecedent part correspond with one of the three combination (B, B, **), (B, **, CE), and (**, B, CE) in the rule-base, we use the product of the confidence degree (as in (5)) of the rule and the degree (as in (6)) of combination as a decision degree. Then select the rule with the maximum decision degree to determine the class label $d^{(i)}$.

For example, if there are only *Rule i* and *Rule j* in the rule-base accord with the combination (B, B, **), *Rule i* is [B, B, S; *Class* 3] with a confidence degree D(*Rule i*) and *Rule j* is [B, B, B; *Class* 2] with a confidence degree D(*Rule j*). The decision degree for *Rule i* is $P_i = 0.54 \times D(\textit{Rule } i)$, and the decision degree for *Rule j* is $P_j = 0.54 \times D(\textit{Rule } i)$. If $P_i \geq P_j$, the class label of the attribute $(c_1^{(i)}, c_2^{(i)}, c_3^{(i)})$ is *Class* 3, otherwise, it is *Class* 2.

There is a small possibility that there might not find any antecedent part in the rule-base corresponding with the combinations of the two attributes either. In this case, we only consider one attribute of $(c_1^{(i)}, c_2^{(i)}, c_3^{(i)})$. These are $(c_1^{(i)}, **, **), (**, c_2^{(i)}, **), (**, **, c_3^{(i)})$, and then, using the method mentioned above to find the class label $d^{(i)}$.

If there could not find any antecedent part in the rule-base corresponding with the combinations of one attribute from the testing sample either, we confirm that it is an outlier and can ignore it, moreover, the probability of this situation approaches to zero as the number of the attributes increasing.

In this way, we get a mapping: $f: \{c_1, c_2, c_3\} \Rightarrow D$.

Our new method can be viewed as a very general model-free data-based fuzzy system. "Model-free" means no mathematical model is required for the problem; "Data-based" means the system constructs directly from "samples" and each sample produces one rule for selection. The system can adaptively change the mapping when new "samples" are available.

TABLE III
DESCRIPTION OF DATA SETS COMING FROM THE UCI REPOSITORY AND USED IN THE EXPERIMENTS

No	Data set	Samples	Attributes	classes
1	Haberman	306	3	2
2	Iris	150	4	3
3	Liver	345	6	2
4	Wine	178	13	3
5	Transfusion	748	4	2
6	Pima	768	8	2
7	Fertility	100	9	2

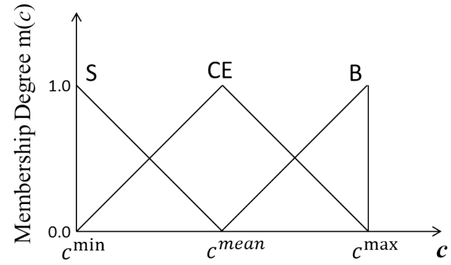


Fig. 3. Fuzzy membership functions of Attribute **c**.

III. EXPERIMENTAL STUDIES

In this section, seven data sets are used in a series of experiments. They come from the UCI Machine Learning Repository. The description of the pertinent data sets is covered in Table III. In the experiments, for each experiment set, a ten-fold cross validation is carried out. Each dataset is stratified and divided into tenfold of (approximately) equal size. Each time one fold is left out of the whole dataset from training, and this fold is used for testing. As a result, there are ten runs for each classifier based on one dataset.

A. Experiment 1

In this experiment, We present a detailed comparative analysis with C4.5 decision tree[23], Decision Table (DTable) [24], OneR [25], Nearest neighbor like Algorithm (NNge) [26], and PART Decision List (PART) [27] over the data sets.

- C4.5 decision tree [23]
  C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set of already classified samples. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the smaller sublists.
- Decision Table [24]
  Decision Table provides a handy and compact way to represent complex business logic. In a decision table, business logic is well divided into conditions, actions (decisions) and rules for representing the various components that form the business logic.
- OneR [25]
  OneR, short for "One Rule", is a simple, yet accurate, classification algorithm that generates one rule for each predictor in the data, then selects the rule with the smallest total error as its "one rule". To create a rule for a predictor, we construct a frequency table for each predictor against the target. It has been shown that OneR produces rules only slightly less accurate than state-of-the-art classification algorithms while producing rules that are simple for humans to interpret.
- Nearest neighbor like Algorithm [26]
  Nearest-neighbor-like algorithm using non-nested generalized exemplars (which are hyperrectangles that can be viewed as if-then rules). The nearest neighbour algorithm was one of the first algorithms used to determine a solution to the traveling salesman problem. In it, the salesman starts at a random city and repeatedly visits the nearest city until all have been visited. It quickly yields a short tour, but usually not the optimal one.
- PART Decision List [27]
  It is applied for generating a PART decision list. Uses separate-and-conquer. Builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule.

The classifiers are implemented by using the Weka machine learning toolkit [29] with the default settings being specified in the toolkit. In my algorithm, we choose the Euclidean distance weight function as the distance function and the parameters are respectively set to be δ = 0.125, and r = 3 suggested by the experiments.

We illustrate the performance of our classifier for the Wine data. The data set consists of 178 samples and each sample is described by thirteen attributes. First the NBASA algorithm in Table I which is proposed in Section II is applied to select an attributes subset with 3 attributes. Then we extract a rule-base from the 178 samples described by the new attribute set and remove the conflicting rules using the method in *step* 3 to get a simplest rule-base. We use membership functions shown in Fig. 3, that is dividing each interval of the three attributes into 3 regions denoted by *S* (Small), *CE* (Center), *B* (Big) and assign each region a triangular fuzzy membership function, The vertex of the *CE* region lies at the mean value of each attribute and the vertices of the other regions, *S* and *B*, lie at the minimum value and maximum value of each attribute regions respectively. After the first simplification in *step* 3, the number of rules in the rule-base is 17.4 (on average), the average classification rate reported for the training set is 82.65% and the average accuracy on the testing set is 89.96%. Then we prune the rule-base by invoking the pruning algorithm shown in TABLE II, the number of rules has been reduced to 9.5 (on average), and for this case, the percentage of correct classification on the training set by the rule-base increased to 84.21% (on average) and the average accuracy on the testing

TABLE IV
THE CLASSIFICATION ACCURACIES OF DIFFERENT CLASSIFICATION METHODS

Data set	C4.5	DTable	OneR	NNge	PART	DBFREM
Haberman	0.70 ± 0.0762	0.72 ± 0.0432	**0.74** ± 0.0558	0.67 ± 0.0738	0.71 ± 0.0688	0.73 ± 0.0563
Iris	0.96 ± 0.0466	0.93 ± 0.0444	0.94 ± 0.0378	0.95 ± 0.0450	0.95 ± 0.0526	**0.97** ± 0.0344
Liver	0.64 ± 0.0748	0.57 ± 0.0571	0.58 ± 0.0849	0.63 ± 0.0851	0.64 ± 0.0878	**0.69** ± 0.0771
Wine	0.92 ± 0.0794	0.92 ± 0.0850	0.78 ± 0.0613	**0.98** ± 0.0300	0.94 ± 0.0546	0.91 ± 0.0650
Transfusion	**0.78** ± 0.0561	0.76 ± 0.0043	0.76 ± 0.0363	0.71 ± 0.0540	**0.78** ± 0.0579	0.77 ± 0.0208
Pima	**0.75** ± 0.0280	**0.75** ± 0.0469	0.71 ± 0.0594	0.73 ± 0.0384	0.73 ± 0.0435	0.75 ± 0.0383
Fertility	0.86 ± 0.0530	**0.88** ± 0.0381	0.87 ± 0.0450	0.82 ± 0.0848	0.86 ± 0.0530	**0.88** ± 0.0381
Average	0.80	0.79	0.77	0.78	0.80	**0.81**

set increased to 91.07%.

The average testing accuracies of the seven data sets are reported in TABLE IV. It shows that the DBFREM can produce high accuracy and it is better than C4.5, DTable, OneR, NNge, and PART on some of the data sets such as Iris, Liver, and Fertility. For Iris data, the average testing accuracy is as high as 97.33%. While, for Liver data, the average accuracy on the testing set of the DBFREM is 68.68%, which are higher than the other five methods.

B. Experiment 2

In this experiment, we compare the number of rules returned by the DBFREM algorithm with other classifiers. By the analysis of the results presented in TABLE V, we can draw the following conclusions:

- The numbers of rules produced by OneR, PART, and DBFREM are of the same order of magnitude on all of the seven datasets and they are less than those produced by the other four methods. The NNge method caused by its algorithm presents the largest numbers of rules almost on all of the dataset in TABLE V.
- The rule-base extracted by DBFREM algorithm is much simpler, easier to understand and easier to interpretable because each antecedent part of the fuzzy rule in the rule-base is extracted from the conjunction of only three single regions.

IV. CONCLUSION

In this paper, we have introduced the neighborhood rough set model as a basic theoretic framework. Based on the neighborhood conception, we developed a NBASA feature selection algorithm to select a small attribute subset from original attribute set. Then we constructed a DBFREM algorithm which extracts a fuzzy rule-base directly from the training samples described by new attribute subset to form a rule-base and then cut down the conflict and redundant rules in the rule-base. The experiments with UCI data sets demonstrated that the obtained results from DBFREM classifier outperform those produced by C4.5, DTable, OneR, NNge, and PART on most of the datasets.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their comments and suggestions.

TABLE V
THE CLASSIFICATION RULES NUMBER OF DIFFERENT CLASSIFICATION METHODS

Data set	C4.5	DTable	OneR	NNge	PART	DBFREM
Haberman	2.90	**1.50**	3.90	76.70	3.40	7.30
Iris	4.70	4.20	**3.00**	9.80	3.50	6.90
Liver	24.90	**2.10**	11.10	101.20	8.00	6.00
Wine	5.30	29.60	**4.70**	9.10	**4.70**	9.50
Transfusion	6.00	**1.00**	3.60	157.30	4.00	3.10
Pima	23.60	32.00	**8.00**	257.20	8.70	8.70
Fertility	1.70	**1.00**	1.10	20.30	5.50	2.00
Average	9.87	10.20	**5.06**	90.23	5.40	6.21

REFERENCES

[1] M. Fazzolari, B. Giglio, R. Alcala, F. Marcelloni, and F. Herrera, “A study on the application of instance selection techniques in genetic fuzzy rule-based classification systems: Accuracy-complexity trade-off”, *Knowledge-Based Systems*, vol. 54, pp. 32-41, 2013.

[2] H. Ishibuchi, T. Nakashima, and M. Nii, “Classification and modeling with linguistic information granules: advanced approaches to linguistic data mining”, Berlin: Springer, 2004.

[3] M. R. Akbarzadeh-Totonchi, and M. Moshtagh-Khorasani, “A hierarchical fuzzy rule-based approach to aphasia diagnosis”, *Journal of Biomedical Informatics*, vol. 40, no. 5, pp. 465-475, 2007.

[4] H. Wu, and J. Mendel, “Classification of battlefield ground vehicles using acoustic attributes and fuzzy logic rule-based classifiers”, *IEEE Transactions on Fuzzy Systems*, vol. 15 no. pp. 56-721, 2007.

[5] J. Marłn-Blzquez, and G. M. Prez, “Intrusion detection using a linguistic hedged fuzzy-XCS classifier system”, *Soft Computing*, vol. 13, no. 3, pp. 273-290, 2009.

[6] C. Tsang, S. Kwong, and H. Wang, “Genetic-fuzzy rule mining approach and evaluation of attribute selection techniques for anomaly intrusion detection”, *Pattern Recognition*, vol. 40,no. 9, pp. 2373-2391, 2007.

[7] S. Abe, and M. S. Lan, “A method for fuzzy rules extraction directly from numerical data and its application to pattern classification”, *IEEE Trans.Fuzzy Systems*, vol. 3, no. 1, pp. 18-28, 1995.

[8] H. Ishibuchi, K. Nozaki, and H. Tanaka, “Distributed representation of fuzzy rules and its application to pattern classification”, *Fuzzy Sets and Systems*, vol. 52, no. 1, pp. 21-32, 1992.

[9] S. Mitra, L. and I. Kuncheva, “Improving classification performance using fuzzy MLP and two-level selective partitioning of the attribute space”, *Fuzzy Sets and Systems*, vol. 70, no. 1, pp. 1-13, 1995.

[10] D. Nauck, and R. Kruse, “A neuro-fuzzy method to learn fuzzy classification rules from data”, *Fuzzy Sets and Systems*, vol. 89, no. 3, pp. 277-288, 1997.

[11] V. Uebele, S. Abe, and M. S. Lan, “A neural-network-based fuzzy classifier”, *IEEE Trans. Systems Man Cybernet*, vol. 25, no, 2, pp. 353-361, 1995.

[12] S. Abe, and R. Thawonmas, “A fuzzy classifier with ellipsoidal regions”, *IEEE Trans. Fuzzy Systems*, vol. 5, no. 3, pp. 358-368, 1997.

[13] J. A. Roubos, M. Setnes, and J. Abonyi, “Learning fuzzy classification rules from labeled data”, *Inform. Sci*, vol. 150, no. 1-2, pp. 77-93, 2003.
[14] J. Casillas, O. Cordn, M. J. Del Jesus, and F. Herrera, “Genetic attribute selection in a fuzzy rule-based classification system learning process for high-dimensional problems”, *Inform. Sci*, vol. 136, no. 1-4, pp. 135-157, 2001.
[15] A. F. Gmez-Skarmeta, M. Valds, F. Jimnez, and J. G. Marłn-Blzquez, “Approximative fuzzy rules approaches for classification with hybrid-GA techniques”, *Inform. Sci*, vol. 136, no. 1-4, pp. 193-214, 2001.
[16] A. Gonzalez, and R. Perez, “SLAVE: a genetic learning system based on an iterative approach”, *IEEE Trans. Fuzzy Systems*, vol. 7, no. 2, pp. 176-191, 1999.
[17] H. Ishibuchi, T. Nakashima, and T. Murata, “Three-objective genetics-based machine learning for linguistic rule extraction”, *Inform. Sci*, vol. 136, no. 1-4, pp. 109-133, 2001.
[18] L. Snchez, I. Couso, J. A. Corrales, O. Cordn, M. J. Del Jesus, and F. Herrera, “Combining GP operators with SA search to evolve fuzzy rule-based classifiers”, *Inform. Sci*, vol. 136, no. 1-4 , pp. 175-191, 2001.
[19] Y. Chung Hu, and G. Hshiung Tzeng, “Elicitation of classification rules by fuzzy data mining”, *Engrg. Appl. of Artificial Intelligence*, vol. 16, no. 7-8, pp. 709-716, 2003.
[20] M. De Cock, C. Cornelis, and E. E. Kerre, “Elicitation of fuzzy association rules from positive and negative examples”, *Fuzzy Sets and Systems*, vol. 149, no. 1, pp. 73-85, 2005.
[21] H. Ishibuchi, and T. Yamamoto, “Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining”, *Fuzzy Sets and Systems*, vol. 141, no. 1, pp. 59-88, 2004.
[22] C. J. Merz, and P. M. Murphy, (1996). UCI Repository for Machine Learning Data-Bases. ([Online]. Available: http://www.ics.uci.edu/ mlearn/MLRepository.html)
[23] J. R. Quinlan, “C4.5: programs for machine learning”, San Francisco: Morgan Kaufmann Publishers, 1993.
[24] R. Kohavi, “The power of decision tables”, *European Conference on Machine Learning*, 1995, pp. 174-189.
[25] R. C. Holte, “Very simple classification rules perform well on most commonly used datasets”, *Machine Learning*, vol. 11, pp. 63-91, 1993.
[26] S. Roy, “Nearest neighbor with generalization”, Christchurch: University of Canterbury, New Zealand, 2002.
[27] E. Frank, and I. H. Witten, “Generating accurate rule sets without global optimization”, *The Fifteenth International Conference on Machine Learning*, 1998, pp. 144-151.
[28] Q. H. Hu, D. R. Yu and Z. X. Xie, “Neighborhood classifiers”, *Expert Systems with Applications*, vol.34, pp. 866-876, 2008.
[29] I. H. Witten, and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed. Morgan Kaufmann, San Mateo, CA, 2005.