Applied Intelligence (2020) 50:2959–2975
https://doi.org/10.1007/s10489-020-01675-7

A novel framework of fuzzy oblique decision tree construction for pattern classification

Yuliang Cai [1] · Huaguang Zhang [1] · Qiang He [2] · Jie Duan [1]

Published online: 19 April 2020

Abstract
In this paper, some significant efforts on fuzzy oblique decision tree (FODT) have been done to improve classification accuracy and decrease tree size. Firstly, to eliminate data redundancy and improve classification efficiency, a forward greedy fast feature selection algorithm based on neighborhood rough set (NRS_FS_FAST) is introduced. Then, a new fuzzy rule generation algorithm (FRGA) is proposed to generate fuzzy rules. These fuzzy rules are used to construct leaf nodes for each class in each layer of the FODT. Different from the traditional axis-parallel decision trees and oblique decision trees, the FODT takes dynamic mining fuzzy rules as decision functions. Moreover, the parameter δ, which can control the size of the tree, is optimized by genetic algorithm. Finally, a series of comparative experiments are carried out with five traditional decision trees (C4.5, Best First Tree (BFT), amulti-class alternating decision tree (LAD), Simple Cart (SC), Naive Bayes Tree (NBT)), and recently proposed decision trees (FRDT, HHCART, and FMMDT-HB) on UCI machine learning datasets. The experimental results demonstrate that the FODT exhibits better performance on classification accuracy and tree size than the chosen benchmarks.

Keywords Fuzzy oblique decision tree · Feature selection · Fuzzy numbers · Fuzzy rule extraction

1 Introduction

Classification and feature selection are important research fields in pattern recognition and machine learning. Classification is a technique modeled by the labeled data and then the label of unlabeled data is identified by this model [1, 2]. Feature selection, as the pre-processing part of the classification technique, can remove redundant features and simplify the construction process of the classifier [3, 4]. Many attribute reduction methods often set the same neighborhood size for all attributes. However, this setting will bring large error due to the fact that there are large differences in the distribution of each attribute data. To handle above issue, a forward greedy fast feature selection algorithm based on neighborhood rough set (NRS_FS_FAST) was proposed [5].

Classification technology based on IF-THEN rules, due to its broad applications, has received increasing interests during the past decades [6, 7]. There are many kinds of classification methods, and decision trees become one of the most well-known classification methods on account of their good learning capability and understanding capability [8–10]. In general, they grow in a top-down way and terminate when all data associated with a node belong to the same class. The existing decision trees can be divided into three types: "standard" decision trees [11–14], fuzzy decision trees [15–18], and oblique decision trees [19–24]. "Standard" decision trees can be used to deal with classification problems. However, they are often not capable of handling uncertainties consistent with human cognitive, such as vagueness and ambiguity. To overcome these deficiencies, fuzzy decision trees have been developed by incorporating the fuzzy uncertainty measure in decision tree construction [16, 25–28]. For example, Liu et al. introduced the coherence membership functions of fuzzy concepts and studied the AFS fuzzy rule-based decision tree classifier [16]. An inductive learning method

✉ Huaguang Zhang
zhanghuaguang@mail.neu.edu.cn

Yuliang Cai
ylcaivv@163.com

Qiang He
heqiangcai@gmail.com

Jie Duan
dj89111@163.com

[1] State Key Laboratory of Synthetical Automation for Process Industries, or College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning, China

[2] College of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning, China

"HAC4.5 fuzzy decision tree" was proposed to obtain a fuzzy decision tree with high predictability by using fuzziness intervals matching with hedge algebra [26]. And a new fuzzy decision tree approach based on Hesitant Fuzzy Sets (HFSs) was introduced to classify highly imbalanced datasets [27]. "Standard" decision trees and fuzzy decision trees are called single variable decision trees (or axis-parallel decision trees) as a result of considering only one attribute at each node to partition the training samples. These decision trees cannot effectively handle these classification problems — the classification boundary is not parallel to axis.

To solve the above-mentioned problem, oblique decision trees were proposed by using the linear combination of all feature components as decision functions [20, 22, 23, 29, 30]. Specifically, a novel bottom-up oblique decision tree induction framework (BUTIF), which did not rely on an impurity-measure for dividing nodes, was proposed in [20]. A new decision tree algorithm, called HHCART, was presented in [22]. It utilized a series of Householder matrices to reflect the training data at each node during the tree construction. The authors in [23] described the application of a differential evolution based approach for inducing oblique decision trees in a recursive partitioning strategy. And the work in [29, 30] adopted geometric structure in the data to assess the hyper planes. However, oblique decision trees were lack of explanation and their computational complexities were high [31].

In order to address the above issues, Wang [17] presented a new architecture of a fuzzy decision tree based on fuzzy rules — fuzzy rule based decision tree (FRDT), which involved multiple features at each node endowed with semantic interpretation. However, some unnecessary fuzzy numbers were considered in fuzzy rules extraction; the same fuzzy numbers were used in all layers of the FRDT, while the data samples on each additional node have changed; and the threshold δ is determined by two-step cross-validation method, of which the search purpose is not clear. Motivated by the above discussions, this paper puts forward a new fuzzy oblique decision tree (FODT), and its growth structure is depicted in Fig. 1.

It stores all training samples at root node. In the first layer of the FODT, the fuzzy rules generated by the FRGA are used to construct leaf nodes as pure as possible. The samples that are not processed by these fuzzy rules are then put into an additional node, which is the only non-leaf node in this layer. If the additional node is not empty and the class number of this node samples is more than two, the FODT continues to grow. In the second layer of the FODT, the fuzzy numbers on additional node are recalculated, and the fuzzy rules generated by the FRGA are used again to construct leaf nodes as pure as possible. Similarly, the samples that cannot be handled by the fuzzy rules in the second layer are then put into a new additional node. Repeating the procedure until the termination condition (the additional node is null, or the samples in the additional node have the same label) is met.

The main contributions of this paper are as follows:

– The NRS_FS_FAST algorithm is introduced to eliminate data redundancy and improve classification efficiency.

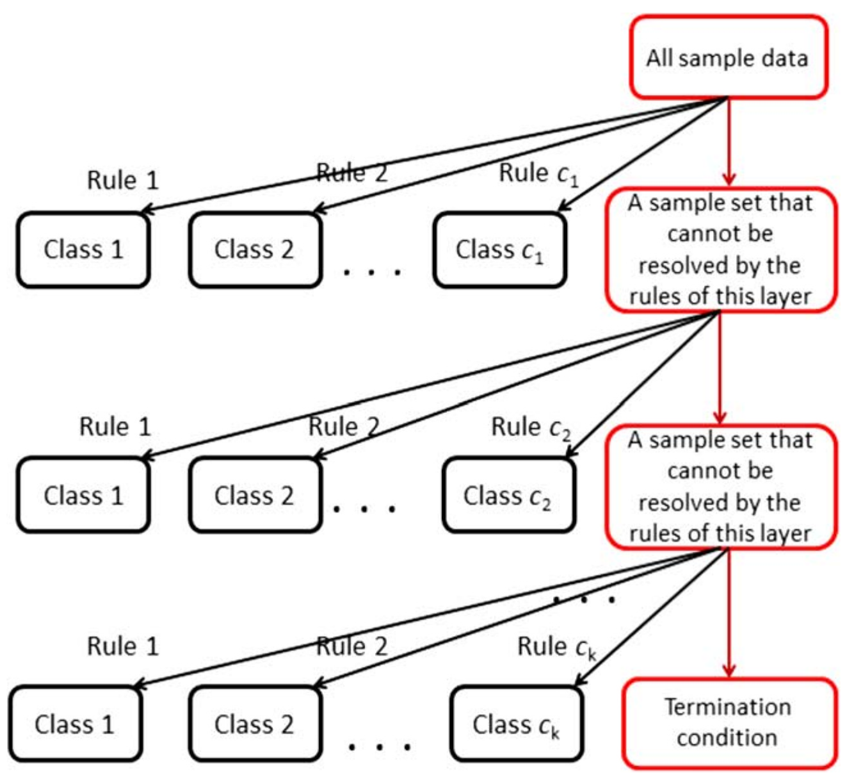


Fig. 1 The growth structure of fuzzy oblique decision tree

- The fuzzy numbers on additional node in each layer are recalculated to obtain fuzzy partition more precisely.
- A new fuzzy rule generation algorithm (FRGA) is put forward to simplify the rules.
- To effectively keep balance between the classification accuracy and tree size, the threshold δ is optimized by the genetic algorithm.

The rest of this paper is organized as follows: the NRS_FS_FAST algorithm and FRGA algorithm are introduced in Section 2. The construction process of the FODT is described in Section 3. In Section 4, several comparative experiments are carried out to verify the effectiveness of the FODT. Section 5 concludes this paper.

2 The extraction of fuzzy rules

Before extracting the fuzzy rules, we first preprocess the raw data, as follows.

2.1 The NRS_FS_FAST algorithm

In recent years, with the development of computer and network technology, a variety of information technologies are widely used in commercial transactions. However, amounts of redundant data might exist in practical applications. Therefore, eliminating these ineffective data and gaining valuable knowledge have become a hot spot.

As we know, attribute reduction is one of the methods for dealing with the problem of data redundancy. Skowron, a famous mathematician in Poland, proposed the method of using discernibility matrix to represent knowledge and then using it to reduce attributes [32], which is simple and easy, but time-consuming. Zhang proposed an efficient heuristic attribute reduction algorithm based on information entropy [33], and Yang proposed an improved heuristic attribute reduction algorithm based on information entropy in rough set [34], which could not only improve the efficiency of attribute reduction, but decrease the number of attribute reduction. To solve inefficiencies, a kind of rough set attribute reduction algorithm was put forward [35] by combining the attribute compatibility model and the attribute importance model. Most of these algorithms set the same neighborhood size for all attributes, which can bring large error if there are large differences in the distribution of each attribute data. In order to reduce these errors, the NRS_FS_FAST algorithm [5] was proposed, which could not only select attributes effectively, but also obtain higher classification accuracy. This paper uses NRS_FS_FAST algorithm to reduce properties, as described in Algorithm 1.

Algorithm 1: Attribute reduction algorithm NRS_FS_FAST(NDT, L).

Input:
NDT : The neighborhood decision system $NDT = \langle U, A, D\rangle$, where U is a set of samples $\{x_1, x_2, x_3, \ldots, x_n\}$, A is a condition attribute set and D is a decision attribute set.
L : The given parameter.
Output:
Red: The attribute subset.

1 **Begin**:
2 $\forall a \in A$, get neighborhood relation matrix: N_a = GetNeighborRelation (NDT, L).
3 Initialize the $red = \varnothing, pos = \varnothing$.
4 red: the attribute reduction set;
5 pos: the positive region of attribute reduction.
6 Select attributes: $Red = SelectAttributes(NDT, L, N_a)$.
7 **for** $i = 1, \ldots, m$ **do**
8 (1) each attribute $a_i \in A - red$;
9 (2) $SIG(a_i, red, D) = \gamma_{red \cup a_i}(D) - \gamma_{red}(D)$;
10 (3) $\underline{N_{red\cup a_i}}(D) = \bigcup_{i=1}^{N} \underline{N_{red\cup a_i}} X_i$. Where $X_1, X_2, \ldots, X_N$ are equivalence classes;
11 (4) Find out the attribute a_k with the largest significance degree and its positive region.
12 $SIG(a_k, red, D) = \max_i\{SIG(a_i, red, D)\}$;
13 $pos = \underline{N_{red\cup a_k}}(D)$.
14 **end**
15 **if** $\Delta(SIG(a_k, red, D)) > 0$ **then**
16 (1) $red = red \cup a_k$;
17 (2) $a_i \in A - red$, delete the pos-th rows and the pos-th columns of N_{a_i};
18 (3) return to line 7.
19 **else**
20 $Red = red$.
21 break.
22 **end**

2.2 Formation of the fuzzy numbers

The training samples are denoted by $X=[x_{ij}]_{n\times m}$, where n is the number of samples, and m is the number of attributes. $x_i=[x_{i1}, x_{i2}, \ldots, x_{im}](i=1,2,\ldots,n)$ represents the i-th sample, $f_j(j=1,2,\ldots,m)$ denotes the j-th attribute of X, $x_{i,j}$ is the j-th attribute value of the i-th sample x_i, $C=\{1,2,\ldots,c\}$ is a set of class labels, where c is the number of classes. X contains c equivalence classes $X_l(l=1,2,\ldots,c)$.

This paper adopts the combination of triangular functions and trapezoidal functions, and stipulates that the number of fuzzy membership functions defined on each attribute is equal to c. For each attribute, the first and the last function are trapezoidal functions, and the rest are triangular functions. Let $f_{j,k}$ denote the k-th$(k=1,2,\ldots,c)$ fuzzy number of the j-th attribute f_j. If the class number $c=4$, the fuzzy membership functions can be depicted as Fig. 2, which shows that the core of fuzzy membership functions lies in the values of parameters $\{ms_{j,k}\}(k=1,2,\ldots,c)$. These values are taken as the mean values of all patterns falling in the k-th equivalence class X_k in this paper. The values of $\left\{ms'_{j,k}\right\}(k=1,2,\ldots,c)$ can be obtained by Eq. (1), and the values of $\{ms_{j,k}\}(k=1,2,\ldots,c)$ are the same values after sorting $\left\{ms'_{j,k}\right\}(k=1,2,\ldots,c)$ in an ascending order.

$$ms'_{j,k}=\frac{\sum_{x_i\in X_k}x_{ij}}{|X_k|}, \tag{1}$$

Example 2 Considering a set of weather data (see Table 1). $X=[x_{i,j}]=\{x_1,\ldots,x_{10}\}$ is a set of 10 samples, $x_i\in R^3$ denotes weather condition of the i-th day, $f_j(j=1,2,3)$ means the tj-th attribute. According to Eq. (1), we can get $ms_{1,1}=(45+48+26+99+69+74)/6=60.1667$, $ms_{1,2}=(42+90+92+73)/4=74.25$, $ms_{2,1}=0.275$, $ms_{2,2}=0.4167$, $ms_{3,1}=2.75$, and $ms_{3,2}=3.667$. Because $c=2$, so K = 2, namely, it only generates two fuzzy numbers on each attribute. The semantics of all fuzzy numbers are $f_{1,1}$: " Low temperature ", $f_{1,2}$: " High temperature ", $f_{2,1}$: " Low humidity ", $f_{2,2}$: " High humidity ", $f_{3,1}$: " Small wind ", and $f_{3,2}$: " High wind " respectively. The membership functions of fuzzy numbers are shown in Fig. 3.

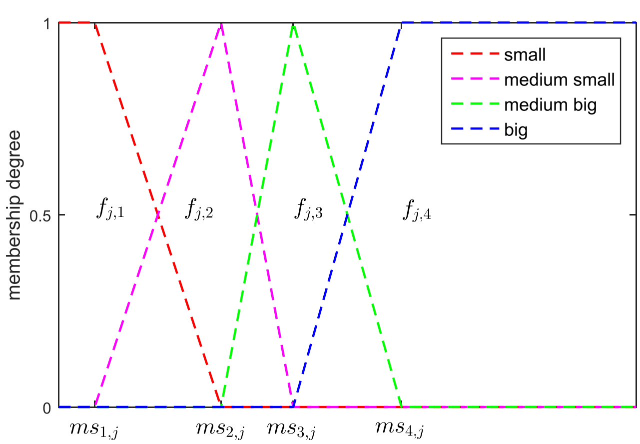


Fig. 2 The triangular and trapezoidal functions on attribute f_j

Table 1 A synthetic weather data

Sample	Temperature	Humidity	Wind	Label
x_1	45	0.7	3	2
x_2	48	0.2	4	2
x_3	42	0.1	1	1
x_4	26	0.1	2	2
x_5	90	0.5	3	1
x_6	99	0.3	3	2
x_7	92	0.2	4	1
x_8	69	0.4	8	2
x_9	72	0.3	3	1
x_{10}	74	0.8	2	2

2.3 Formation of fuzzy rules

Fuzzy IF-THEN rules can be described as follows [36]:

> R: IF the value of x_{i1} is small and the value of x_{i2} is big, THEN x_i belongs to class 1.

Taking the fuzzy numbers defined in Section 2.2 into account, the fuzzy IF-THEN rule can be rewritten as follows:

> R: IF x_i is $f_{1,1}$ and $f_{2,2}$, THEN x_i belongs to class 1.

For each fuzzy set $A\subseteq F$, $\prod_{f\in A}$ represents the conjunction of fuzzy numbers in A. For instance, $A=\{f_{1,1},f_{2,2}\}\subseteq F$, $\prod_{f\in A}f=f_{1,1}f_{2,2}$. Therefore, the fuzzy rule can be further rewritten as:

> R: IF x_i is $f_{1,1}f_{2,2}$, THEN x_i belongs to class 1.

2.4 The fuzzy rule generation algorithm (FRGA)

Fuzzy IF-THEN rules play a vital role in constructing the FODT. A classical association between properties A and B can be described as $A\Rightarrow B$, which indicates that an element satisfying property A is also able to satisfy property titB. Two indices (*Support* and *Confidence* [37]) are often used to measure the validity of such association rule. The support degree is defined as $Supp(A\Rightarrow B)=|A\cap B|/|X|$, and the confidence degree is defined as $Conf(A\Rightarrow B)=|A\cap B|/|A|$. When applying fuzzy rules to study the classification problem, we can generalize two measures: fuzzy support degree (*FSupp*) and fuzzy confidence degree (*FConf*) [38]. Specific formulas are as follows:

$$FSupp(A \Rightarrow l) = \frac{\sum_{x \in X_l} A(x)}{|X|}, \tag{2}$$

$$FConf(A \Rightarrow l) = \frac{\sum_{x \in X_l} A(x)}{\sum_{x \in X} A(x)}, \tag{3}$$

$$A(x) = \frac{\sum_{f \in A} f(x)}{|A|}, \tag{4}$$

where A indicates $\prod_{f \in A} f$, $A(x)$ is the average membership degree that x belongs to the fuzzy set $\prod_{f \in A} f$. $|\cdot|$ represents the number of elements over a set. When confirming the fuzzy number with the maximum fuzzy confidence degree, we determine the corresponding attribute of this fuzzy number and then remove the remaining fuzzy numbers concerning this attribute to reduce unnecessary fuzzy numbers. The specific steps of the FRGA are shown in Algorithm 2.

3 The construction of the FODT

3.1 The theory and algorithm of the FODT

The architecture of the FODT is developed by using fuzzy "if-then" rules, shown in Fig. 4.

Firstly, putting all training data X into the root node of the tree. The fuzzy rules extracted by the FRGA are denoted as $R_{1,\ l}(l = 1, \ldots, c_1)$, where c_1 represents the class number of X. Only one rule is extracted for each class, so $c_1 = c$. The subscript "1" of the rule $R_{1,\ l}$ indicates the first layer of the tree, the subscript "l" indicates the l-th class. Each class is assigned a leaf node, which contains as many samples that belongs to this class as possible. The samples that can not be contained in these leaf nodes are put into an additional node $X^{1,\ \delta}$, which is the only non-leaf node in first layer.

Algorithm 2: The fuzzy rule generation algorithm (FRGA).

Input:
$X = [x_{i,j}]_{n \times m}$: A set of training samples which consists of c classes $X_l(l = 1, \ldots, c)$;
$M = \{1, 2, \ldots, m\}$;
β: The parameter used to adjust the number of fuzzy numbers in the corresponding rule and its fuzzy confidence degree $FConf$ ($0 \leq \beta \leq 1$, and usually is set to 0.02);
H : The parameter of controlling the number of fuzzy numbers in the corresponding rule.
Output:
R_l : The obtained fuzzy rule to describe the class X_l.

1 **Begin**:
2 Initialize t=1, $f_m = \varnothing$, and $f_t = \varnothing$.
3 t: the parameter that controls cycle times;
4 f_m: the fuzzy number contained in the fuzzy rule R_l.
5 Calculate the fuzzy numbers $f_{j,k}(j \in M, k = 1, 2, \ldots, c)$ of non-leaf node samples. $F = \{f_{j,k}\}(j \in M, k = 1, 2, \ldots, c)$ is a set of fuzzy concepts.
6 **for** l=1,...,c **do**
7 Calculate the fuzzy confidence degrees $FConf(f \Rightarrow l)$ of all fuzzy numbers:
8 $\forall f \in F$, compute $FConf(f \Rightarrow l)$.
9 Determine the corresponding fuzzy number with the maximum fuzzy confidence degree:
10 $f_{a_t,b_t} = arg\ max_f FConf(f_{j,k} \Rightarrow l)(j \in M, k = 1, 2, \ldots, c)$;
11 $f_m = \{f_{a_t,b_t}\}$. $f_t = \{f_{a_t,b_t}\}$. $FC_t = FConf(f_t \Rightarrow l) + t * \beta$.
12 Remove the fuzzy numbers $\{f_{a_t,k}\}(k = 1, 2, \ldots, c)$ of the a_t-th attribute and obtain the rest attributes:
13 $M = M \setminus a_t$.
14 The remaining fuzzy numbers are denoted as:
15 $F_M = \{f_{j,k}\}(j \in M, k = 1, 2, \ldots, c)$.
16 **if** $M \neq \varnothing$ *and the length of* $f_m < min(H, c * m)$ **then**
17 $\forall f \in F_M$, compute $FConf(f \wedge f_t \Rightarrow l)$;
18 $f_{a_{t+1},b_{t+1}} = arg\ max_f FConf(f \wedge f_t \Rightarrow l)$;
19 $f_m = f_m \vee f_{a_{t+1},b_{t+1}}$, $f_{t+1} = f_t \wedge f_{a_{t+1},b_{t+1}}$;
20 $FC_{t+1} = FConf(f_{t+1} \Rightarrow l) + (t+1) * \beta$;
21 $t = t + 1$.
22 **return** to line 12.
23 **else**
24 $d = arg\ max_t FC_t$, so the antecedent part of the fuzzy rule R_l is f_d.
25 break.
26 **return** R_l.
27 **end**
28 **end**

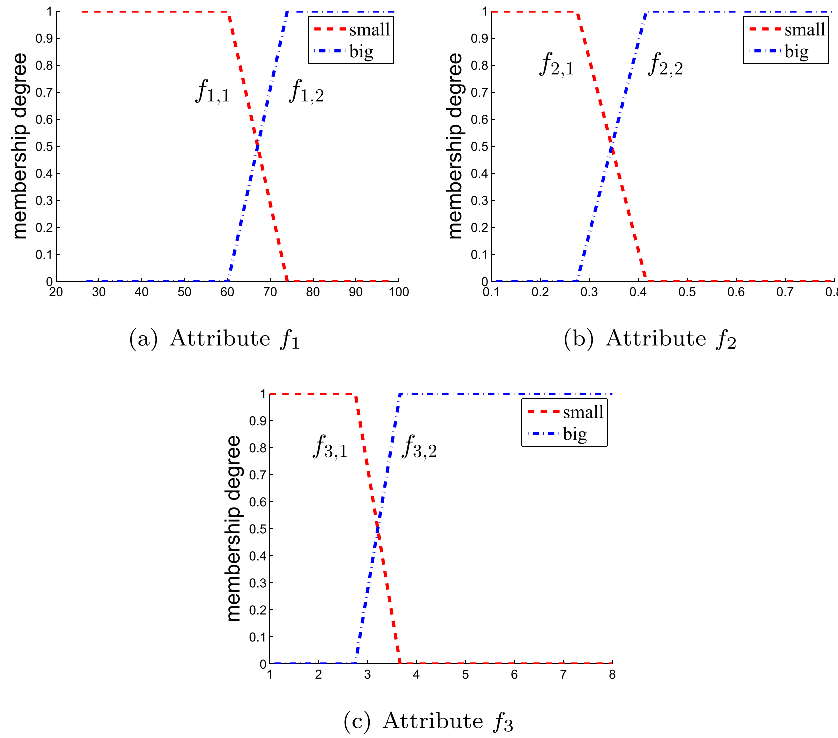


Fig. 3 The membership functions of fuzzy numbers

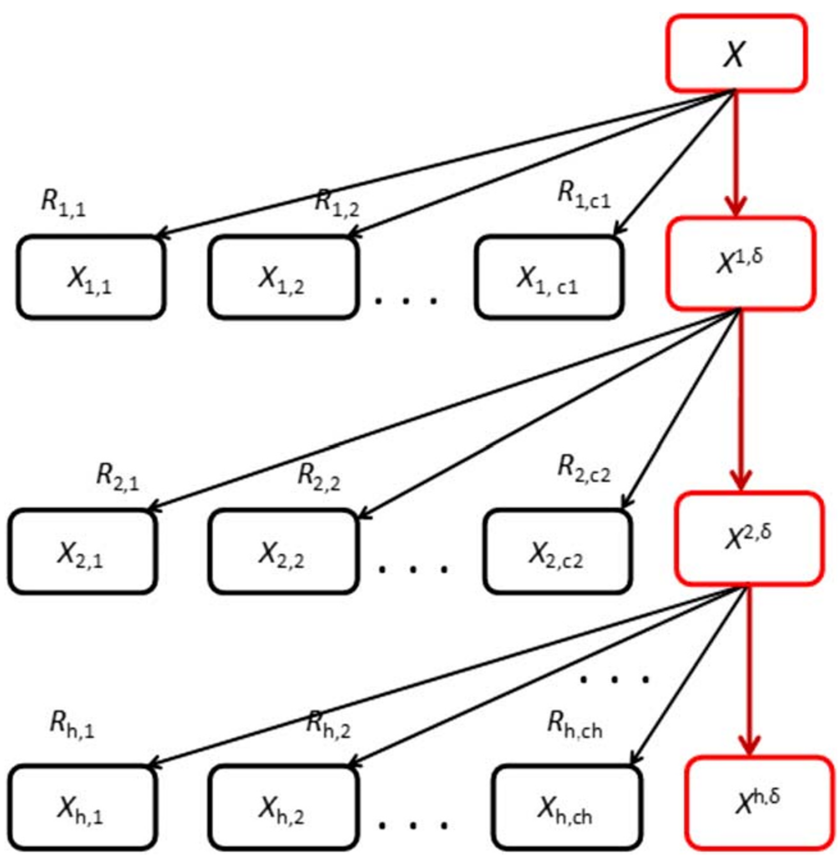


Fig. 4 The overall structure of the FODT

Secondly, the FODT continues to grow on additional node $X^{1,\ \delta}$. The fuzzy numbers of $X^{1,\ \delta}$ are recalculated, and the FRGA based on these new fuzzy numbers is adopted again to extract fuzzy rules for each class contained in $X^{1,\ \delta}$, denoted by $R_{2,\ l}(l = 1, \ldots, c_2)$ respectively, where c_2 represents the class number of $X^{1,\ \delta}$. And then we put the samples that cannot be determined by the second layer rules $R_{2,\ l}(l = 1, \ldots, c_2)$ into an additional node $X^{2,\ \delta}$, which is also an only non-leaf node in second layer.

$\ddots$

Finally, the FODT continues to grow on additional node $X^{h,\ \delta}$ until one of the termination conditions is satisfied:

(i) $X^{h,\ \delta} = \varnothing$, which means that all samples have been determined by the rules $R_{h,\ l}(l = 1, \ldots, c_h)$.
(ii) $X^{h,\ \delta} = X^{h-1,\ \delta}$, which means that the rules in the h-th layer lose its effect.
(iii) $c_h = 1$, which means that the samples in the h-th layer have the same class label.

Definition 1 Assuming the antecedent part of the rule $R_{h,\ l}$ is $A_{h,\ l}$, where $A_{h,\ l}(x)$ denotes the average membership degree of the sample x belonging to the fuzzy concepts contained in $A_{h,\ l}$. The non-leaf node in the h-th layer is defined as $X^{h,\delta} = X^{h-1,\delta}\left(R_{h,1}, R_{h,2}, \ldots, R_{h,c_h}, \delta\right)$, which satisfies:

(i) $X^{h,\ \delta} \subseteq X^{h-1,\ \delta}$.
(ii) $\forall x \in X^{h,\ \delta}$, $\forall l \in 1, 2, \ldots, c_h$, $A_{h,\ l}(x) < \delta$.

The FODT algorithm is shown in Algorithm 3. After constructing the FODT, each leaf node corresponds to a fuzzy IF-THEN rule.

3.2 Determination of the optimal threshold δ

The growth process and tree structure of the FODT can be controlled by the threshold δ. If the given threshold δ is small, we will get a larger decision tree, and if δ is large, the decision tree will be small. Obviously, the classification results of the fuzzy decision tree greatly depends on δ. However, it is difficult to directly obtain the optimal threshold δ from the training data. In this paper, we construct an objective function and use genetic algorithm to optimize δ, as follows:

$$F(\delta) = |X|*C_a - \delta*N, \tag{5}$$

where $|X|$ is the number of training samples, C_a is the classification accuracy of the training samples, and N is the number of leaf nodes.

Algorithm 3: The FODT algorithm.

Input:
X: A set of training samples which consists of c classes $X_l(l = 1, \ldots, c)$;
$MaxL$: The maximum length of the fuzzy rules;
β : A penalty factor used to control the length of the fuzzy rules;
δ : The margin value, $0 \le \delta \le 1$.
Output:
$R_{h,l}$: The obtained fuzzy rule;
$X_{h,l}$: The training samples can be decided by the fuzzy rule $R_{h,l}$;
$X^{h,\delta}$: The samples on non-leaf node.

Begin:
  $X^{0,\delta} = X$, h=1.
  while $X^{h-1,\delta} \neq \varnothing$ **do**
    Compute the classes number c_h of $X^{h-1,\delta}$.
    if $c_h = 1$ **then**
      break.
    end
    Compute $R_{h,l} = FRGA(X^{h-1,\delta}, MaxL, \beta)(l = 1, \ldots, c_h)$.
    Compute $X^{h,\delta} = X^{h-1,\delta}(R_{h,1}, R_{h,2}, \ldots, R_{h,c_h}, \delta)$.
    if $X^{h,\delta} = X^{h-1,\delta}$ *or* $X^{h,\delta} = \varnothing$ **then**
      break.
    end
    $h = h + 1$.
    return $R_{h,l}, X_{h,l}$, and $X^{h,\delta}$.
end

3.3 Rules of the FODT

Let x be a test sample, h represent the h-th layer of the FODT, l indicate the l-th class. $A_{h,\,l}$ denotes the antecedent part of the fuzzy rule $R_{h,\,l}$ and c_h represents the number of class in the h-th layer. The specific rules of the FODT are as follows:

IF x is $A_{1,1}$, THEN class 1
ELSE IF x is $A_{1,2}$, THEN class 2
$\ddots$
ELSE IF x is A_{1,c_1}, THEN class c_1
ELSE
IF x is $A_{2,1}$, THEN class 1
ELSE IF x is $A_{2,2}$, THEN class 2
$\ddots$
ELSE IF x is A_{2,c_2}, THEN class c_2
ELSE
$\ddots$
IF x is $A_{h,1}$, THEN class 1
ELSE IF x is $A_{h,2}$, THEN class 2
$\ddots$
ELSE IF x is A_{h,c_h}, THEN class c_h
ELSE
...

3.4 Analysis of the time complexity

In this section, we study the time complexity of the FODT. Assuming that the FODT is built on one dataset with n training instances, m attributes, and c class label. The FODT is composed of two main phases: NRS_FS_FAST and FRGA. However, the major computation cost is used over the second phase. For the FRGA phase, the maximum length of fuzzy rules is H, thus the time complexity of FRGA is $O(H*c*n)$. The number of layers of FODT, in the worst situation, is the number of samples, i.e., only one sample of training data is determined on each layer of the tree. Therefore, the time complexity of FODT is $O(H*c*n*n)$ in the worst situation. Moreover, the depth of the tree is on the order of $O(logn)$. Thus, the total time complexity of the FODT is $O(H*c*n*logn)$.

4 Experimental results and analysis

In this section, we evaluate the performance of the FODT in several experiments. To verify the effectiveness of the FODT, we compare our method with some well-known decision tree algorithms, such as "traditional" decision tree algorithms (C4.5 [11], BFT [39], LAD [40], SC [41], and NBT [42]) and recently proposed decision tree algorithms (FRDT [17], HHCART [22], and FMMDT-HB [43]). In addition, we conduct experiments with ten-times ten-folds cross validation on twenty datasets acquired from the UCI Machine Learning repository [44] and one biomedical dataset [45]. The features of these datasets are given in Table 2. In the experiment, the parameter L in the NRS_FS_FAST is set to 2, and the adjustment parameters H and β in the FRGA are set to 5 and 0.02 respectively. Moreover, for all the traditional algorithms, we use their Weka implementation [46], and the values of the parameters for all the traditional algorithms are set to their default values.

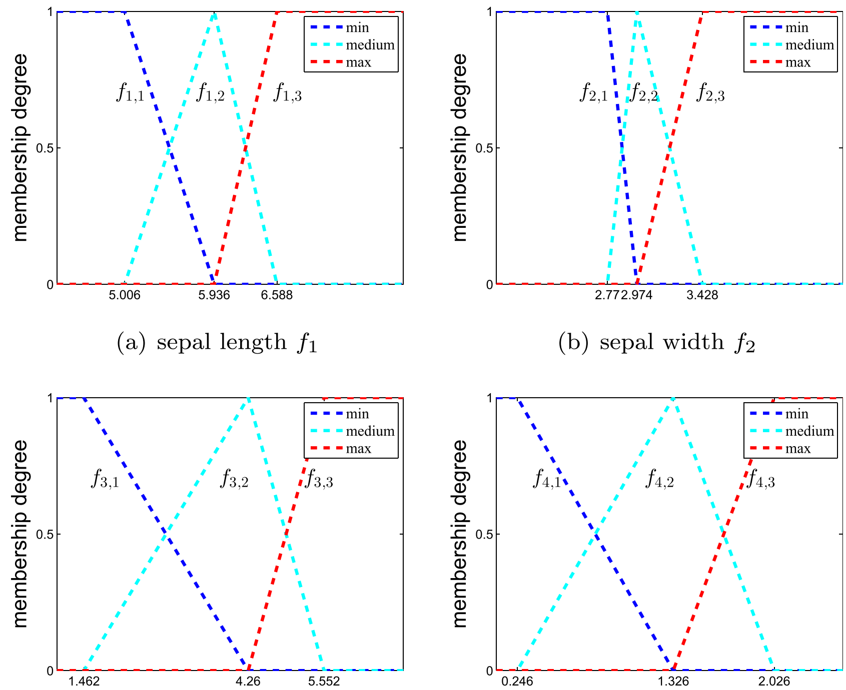
4.1 The experiment on iris data

We illustrate the performance of the FODT on Iris data. The iris data can be described by $X=[x_{i,\,j}]_{150\times 4}$ with three species: iris-setosa (class 1), iris-versicolor (class 2) and iris-virginica (class 3), and the data contains four attributes: sepal length (f_1), sepal width (f_2), petal length (f_3), and petal width (f_4). $x_i=[x_{i,\,1}, x_{i,\,2}, x_{i,\,3}, x_{i,\,4}]$ $(1\le i\le 150)$ represents the i-th sample.

The NRS_FS_FAST algorithm in Table 1 is firstly used to reduce attributes. Iris data is not reduced by the NRS_FS_FAST algorithm, i.e., we get all attributes of iris data after NRS_FS_FAST algorithm.

Table 2 Description of the experimental datasets

No	Dataset	Sample	Attribute	Class
1	Iris	150	4	3
2	Wine	178	13	3
3	Wdbc	569	30	2
4	Credit	690	14	2
5	Heart	270	13	2
6	Haberman	306	3	2
7	Newthyroid	215	5	3
8	Wobc	699	9	2
9	Column_3C	310	6	3
10	Breast Cancer	638	9	2
11	Ionosphere	351	34	2
12	LiverDisorder	345	6	2
13	Sonar	208	60	2
14	Vehicle	846	18	4
15	Boston Housing	506	13	2
16	BUPA	345	6	2
17	Pima Indian	768	8	2
18	Survival	306	3	2
19	Waveform1	5000	21	3
20	Waveform2	5000	40	3
21	ALLAML	72	7129	2



(a) sepal length f_1 (b) sepal width f_2

(c) petal length f_3 (d) petal width f_4

Fig. 5 Membership functions of fuzzy numbers on iris data

Next, the fuzzy numbers $F = \{f_{j,\ k} | 1 \leq j \leq 4, 1 \leq k \leq 3\}$ are obtained by the definition of fuzzy numbers in Section 2.2. And the semantics of the fuzzy numbers are as follows: $f_{1,\ 1}$: " The length of sepal is short ", $f_{1,\ 2}$: " The length of sepal is medium ", $f_{1,\ 3}$: " The length of sepal is long "; $f_{2,\ 1}$: " The width of sepal is short ", $f_{2,\ 2}$: " The width of sepal is medium ", $f_{2,\ 3}$: " The width of sepal is long "; $f_{3,\ 1}$: " The length of petal is short ", $f_{3,\ 2}$: " The length of petal is medium ", $f_{3,\ 3}$: " The length of petal is long "; $f_{4,\ 1}$: " The width of petal is short ", $f_{4,\ 2}$: " The width of petal is medium ", and $f_{4,\ 3}$: " The width of petal is long ". The fuzzy membership functions of these fuzzy numbers are shown in Fig. 5. Given parameter δ =0.5, the FODT is shown in Fig. 6. And the corresponding rules are as follows:

IF x is $f_{1,\ 1}f_{2,\ 1}$, THEN class 1
ELSE IF x is $f_{1,\ 2}f_{2,\ 2}$, THEN class 2
ELSE IF x is $f_{1,\ 3}f_{2,\ 3}$, THEN class 3
ELSE x belong to class 3.

The semantics of the fuzzy rules are "the samples which the petal length is short and the petal width is short belong to class 1; the samples which the petal length is medium and the petal

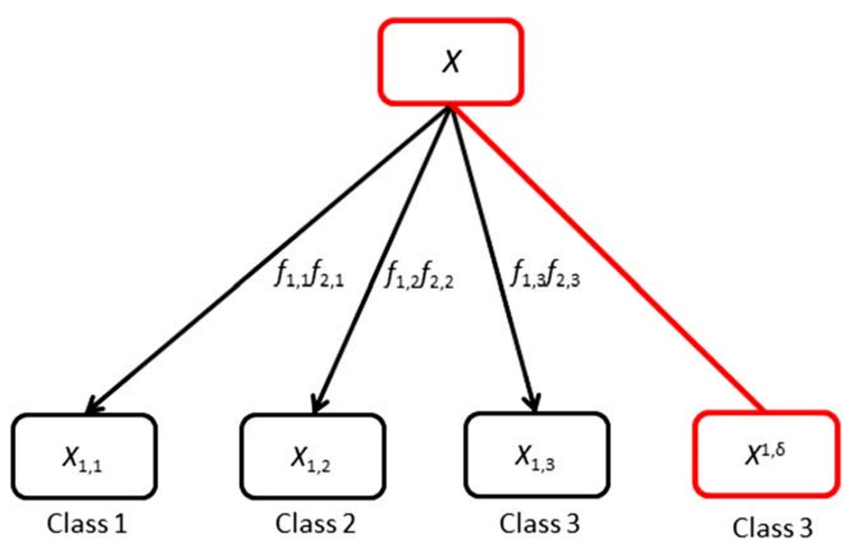


Fig. 6 The structure of the FODT on Iris data with $\delta = 0.5$, $H = 5$

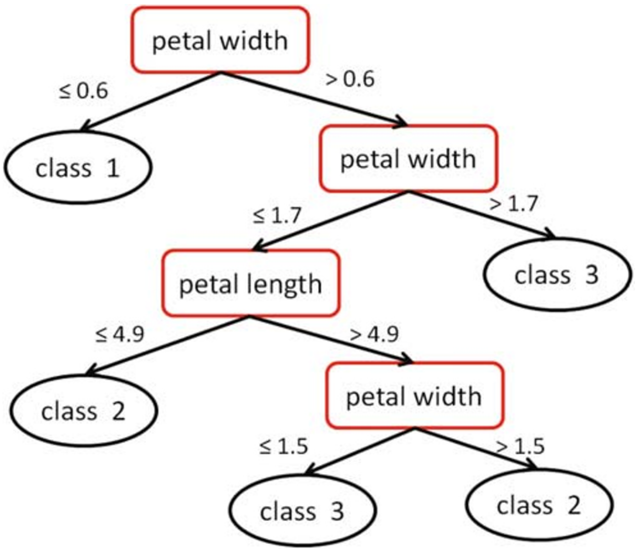


Fig. 7 The tree obtained by C4.5 on Iris data

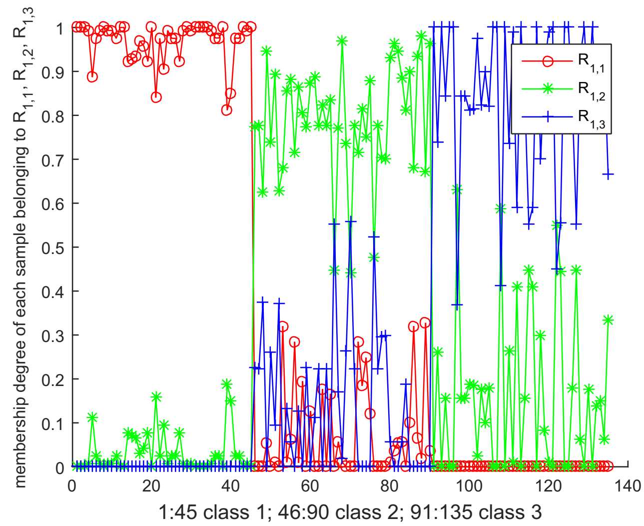


Fig. 9 The membership degrees of Iris training data belonging to fuzzy rules $R_{1,\ 1}$, $R_{1,\ 2}$ and $R_{1,\ 3}$

width is medium belong to class 2; and the samples which the petal length is long and the petal width is long belong to class 3". It can be seen that the rules obtained by the FRGA are easy to understand. Figure 7 shows the C4.5 decision tree. From Figs. 6 and 7, we can see that the tree structure obtained by the FODT is more concise, that is, the rules of the FODT are obviously less than that of the C4.5 tree.

Moreover, the three-dimensional classification result of Iris training data by fuzzy rules $R_{1,\ 1}$, $R_{1,\ 2}$, and $R_{1,\ 3}$ is shown in Fig. 8. The red circles indicate the samples determined by the rule $R_{1,\ 1}$, the green squares representthe samples determined by the rule $R_{1,\ 2}$, and the blue diamonds stand for the samples determined by the rule $R_{1,\ 3}$. Besides, arrow 1 indicates that the rule $R_{1,\ 2}$ divides the samples that belong to the third class into the second class, andarrow 2 represents that the rule $R_{1,\ 3}$ divides the samples that belong to the second class into the third class. The membership degrees of Iris training data belonging to fuzzy rules $R_{1,\ 1}$, $R_{1,\ 2}$ and $R_{1,\ 3}$ are depicted in Fig. 9. It shows that the samples in the first class originally belong to the rule $R_{1,\ 1}$ with the largest membership degrees, the samples in the second class originally belong to the rule $R_{1,\ 2}$ with the largest membership degrees, and the samples in the third class originally belong to the rule $R_{1,\ 3}$ with the largest membership degrees. That is, we can obtain satisfactory results by using the fuzzy rules $R_{1,\ 1}$, $R_{1,\ 2}$ and $R_{1,\ 3}$ to clarify the iris dataset.

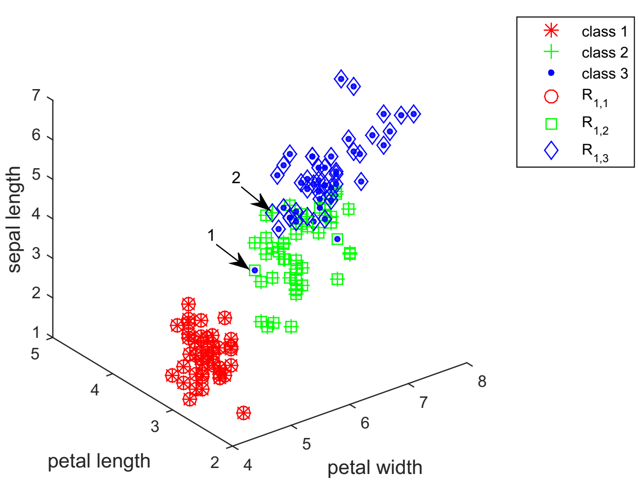


Fig. 8 The three-dimensional classification result of Iris training data by fuzzy rules $R_{1,\ 1}$, $R_{1,\ 2}$ and $R_{1,\ 3}$

4.2 Comparison of the FODT with FMMDT-HB

FMMDT-HB [43] was a decision tree learning algorithm proposed by Mirzamome and Kangavar in 2017. In this subsection,

Table 3 The number of MI and TS along with the respective standard deviations of FMMDT-HB and FODT, and the best scores are indicated in boldface

Dataset	NMI		TS	
	FMMDT-HB	FODT	FMMDT-HB	FODT
Breast Cancer	**31.9** ± 5.8	35.1 ± 3.1	13.0 ± 1.5	**5.3** ± 1.0
Column_3C	54.1 ± 4.0	**43.7** ± 2.1	17.0 ± 0.0	**7.3** ± 0.3
Haberman	**76.9** ± 8.7	77.1 ± 2.8	20.6 ± 2.2	**4.0** ± 0.7
Ionosphere	64.9 ± 2.4	**37.4** ± 1.7	**4.0** ± 0.0	7.7 ± 0.9
Iris	7.8 ± 2.0	**5.8** ± 0.8	10.0 ± 1.1	**4.1** ± 0.1
LiverDisorder	123.7 ± 8.8	**111.4** ± 3.7	22.0 ± 0.1	**12.5** ± 2.3
Newthyroid	11.1 ± 2.9	**7.0** ± 0.7	11.1 ± 2.0	**5.8** ± 0.4
Sonar	63.5 ± 8.1	**48.0** ± 2.4	18.9 ± 1.7	**6.5** ± 0.8
Vehicle	402.0 ± 16.5	**351.3** ± 18.4	31.1 ± 0.3	**23.7** ± 2.1
Average	92.88 ± 6.58	**79.64** ± **3.97**	16.41 ± 0.99	**8.54** ± **0.96**

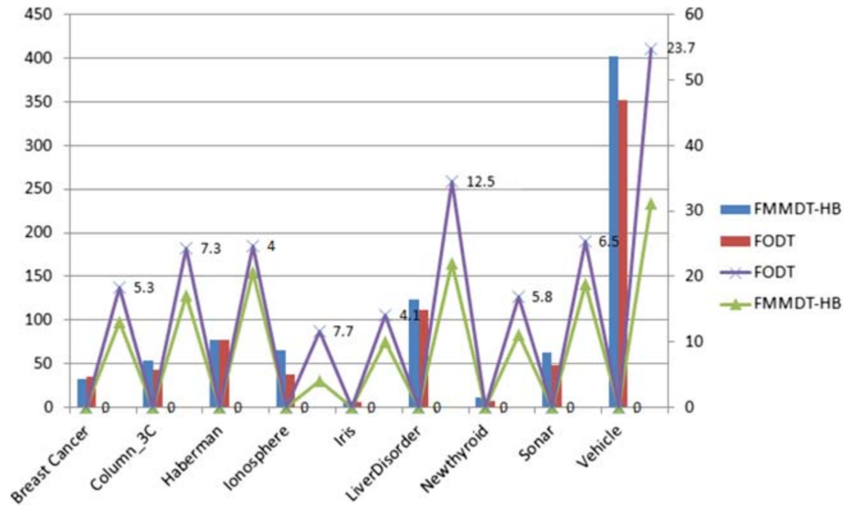


Fig. 10 The number of MI and TS along with the respective standard deviations of FMMDT-HB and FODT

we compare the presented algorithm with FMMDT-HB regarding both the classification accuracy and tree size. For better comparison, we use the datasets in [43], and the parameter settings of FMMDT-HB are consistent with [43].

Table 3 shows the number of misclassified instances (MI) and tree sizes (TS) along with the respective standard deviations of the FMMDT-HB and FODT. The results are the average results over ten runs of ten-folds cross validation. It can be observed that for most of the datasets, the FODT presents the less number of MI and TS. Specifically, the FODT, in the number of MI, is less than FMMDT-HB tested for the all datasets except Breast Cancer and Haberman. Besides, the decision trees generated by FODT are much smaller than those generated by FMMDT-HB, and FODT outperforms FMMDT-HB with eight of the nine datasets. We can conclude that the performance of the FODT has been verified. Moreover, Fig. 10 depicts these results. Compared with these results, we can see that the FODT achieves the better performance than the rival algorithm.

4.3 Comparison of the FODT with HHCAT

HHCART [22] was a well-known oblique decision tree proposed in 2016. In this subsection, we compare FODT with HHCART in terms of classification accuracy, tree size, and time complexity. Similarly, the datasets in [22] are applied to carry out thecomparative experiments.

From the work in [22], the time complexities of HHCART(A) and HHCART(D) are $O(c * n^2 * m^3)$ and $O(c * n^2 * m^2)$ respectively. The time complexity of FODT is $O(H * c * n * logn)$ obtained from Section 3.4. Obviously, the time complexity of HHCART(A) is higher than that of HHCART(D). Therefore, we only need to compare the time complexity of HHCART(D) and FODT. In general, $logn < n$, $m^2 > H = 5$, thus, it can be concluded that the time complexity of FODT is the lowest compared with the chosen benchmarks.

Table 4 shows the detailed comparison results in terms of the number of MI and TS along with the respective standard deviations of the HHCART and FODT. It is clear that the accuracy

Table 4 The number of MI and TS along with the respective standard deviations of HHCART(A), HHCART(D) and FODT, and the best scores are indicated in boldface

Dataset	NMI			TS		
	HHCART(A)	HHCART(D)	FODT	HHCART(A)	HHCART(D)	FODT
Boston Housing	84.5 ± 4.6	86.0 ± 3.5	**68.9** ± 2.3	**6.5** ± 2.1	9.9 ± 2.6	16.7 ± 1.3
Breast Cancer	**19.1** ± 1.9	**19.1** ± 1.9	35.1 ± 3.1	**2.4** ± 0.6	2.6 ± 1.1	5.3 ± 1.0
BUPA	123.9 ± 9.0	129.7 ± 8.6	**111.0** ± 6.5	**6.5** ± 1.5	8.6 ± 3.1	12.8 ± 2.0
Heart	69.9 ± 7.8	65.3 ± 7.6	**38.7** ± 2.1	**4.5** ± 1.7	7.8 ± 2.6	5.6 ± 1.3
Pima Indian	213.5 ± 15.4	208.1 ± 10.0	**187.0** ± 5.1	9.1 ± 5.1	10.8 ± 4.4	**8.3** ± 2.5
Wine	15.5 ± 2.8	20.1 ± 5.5	**7.7** ± 1.1	**3.4** ± 0.3	4.5 ± 0.6	4.1 ± 0.1
Survival	81.1 ± 4.6	83.2 ± 3.1	**78.3** ± 2.4	5.3 ± 2.7	5.0 ± 2.4	**4.1** ± 0.4
Average	86.79 ± 6.59	87.36 ± 5.74	**75.24** ± **3.23**	**5.39** ± 2.00	7.03 ± 2.40	8.13 ± **1.23**

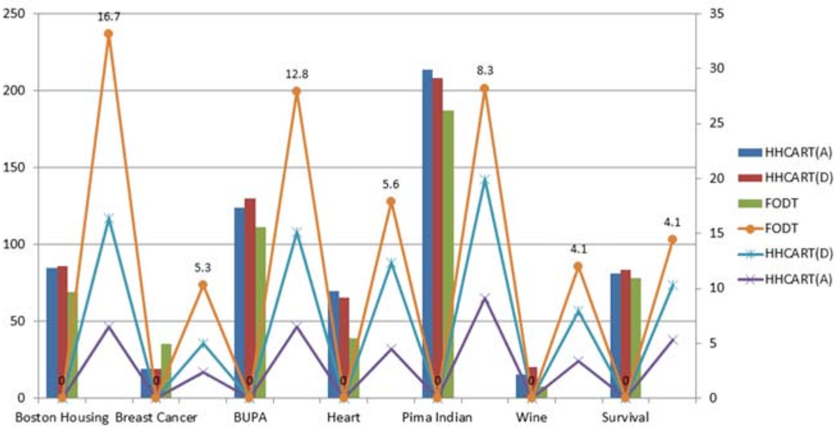


Fig. 11 The number of MI and TS along with the respective standard deviations of HHCART(A), HHCART(D) and FODT

of our method is significantly higher than those of the HHCART(A) and HHCART(D) for all datasets except Breast Cancer. Besides, for each dataset, FODT produces fewer leaf nodes than HHCART(D), especially in datasets "Heart", "Pima Indian", "Wine" and "Survival". It also denotes that the average tree size of the FODT is only 1.1 more than that of HHCAT(D). It is worth mentioning that, although the average number of TS in this paper is more than HHCART(A) and HHCART(D), the performances of classification accuracy and time complexity are better than those of the comparison algorithms. Moreover, the results in Table 4 are also depicted vividly in Fig. 11. These results advocate the superiority of FODT in producing more accurate decision trees.

4.4 Comparison of the FODT with five conventional decision trees and FRDT

4.4.1 Comparison on number of MI and TS

In order to better demonstrate the superiority of the FODT, this paper compares our method with its state-of-the-art competitors: C4.5 [11], BFT [39], LAD [40], SC [41], NBT [42] and FRDT [17].

Table 5 summarizes the results on the number of MI of the FODT and the chosen benchmarks. The average number of MI by ten-times ten-folds cross classifications are listed in Table 5. The notation "$*$" here indicates that the NBT algorithm is ineffective for ALLAML dataset. Compared with six state-of-the-art methods, our method obtains the highest accuracy in all datasets than decision trees: SC, BFT, LAD, and SC. It is not as good as the C4.5 onlyin one dataset (Wobc) and FRDT only in two datasets (Wobc and Waveform1). These results indicate that the FODT provides higher classification accuracy compared with the rival algorithms.

The number of TS and standard deviation for the FODT and the chosen benchmarks is showed in Table 6. The notation "$*$" here share the same meaning as Table 5. It is clear that the scale of the FODT isnot as good as the traditional decision trees (LAD, BFT, C4.5 and NBT) only in one dataset and SC only in two datasets. Tables 5 and 6 also show that the average number of MI and TS of the FODT is significantly less than the chosen benchmarks. Therefore, we can conclude that the FODT can generate more accurate and simpler decision trees. To better demonstrate the superiority of our approach, we draw a bar graph in Figs. 12 and 13. Obviously, our method has the better performance on both the classification accuracy and the size of tree.

Table 5 The number of MI and standard deviation for the FODT and the chosen benchmarks, and the best scores are indicated in boldface

Dataset	BFT	C4.5	LAD	SC	NBT	FRDT	FODT
Iris	8.4 ± 1.3	7.9 ± 1.2	8.3 ± 1.3	8.7 ± 1.5	9.8 ± 1.9	6.2 ± 0.6	**5.8** ± 0.8
Wine	18.6 ± 2.2	12.1 ± 2.4	23.0 ± 3.2	18.7 ± 3.0	**7.0** ± 2.4	10.8 ± 1.5	7.7 ± 1.1
Wdbc	39.6 ± 3.7	35.5 ± 2.8	53.2 ± 6.9	38.9 ± 3.5	34.4 ± 4.6	28.2 ± 1.7	**18.0** ± 2.5
Credit	106.2 ± 3.6	111.0 ± 4.9	141.6 ± 12.8	105.7 ± 4.5	109.2 ± 4.1	102.3 ± 4.8	**81.9** ± 3.1
Heart	61.5 ± 4.6	59.0 ± 6.1	74.6 ± 5.2	59.2 ± 4.4	51.5 ± 3.3	44.6 ± 3.0	**38.7** ± 2.1
Haberman	84.4 ± 4.5	85.2 ± 3.4	90.2 ± 7.4	81.9 ± 3.7	87.0 ± 4.0	81.2 ± 2.3	**77.1** ± 2.8
Newthyroid	15.2 ± 1.7	15.9 ± 2.1	23.9 ± 2.2	17.5 ± 2.0	16.4 ± 3.0	14.4 ± 3.1	**7.0** ± 0.7
Wobc	38.8 ± 3.8	34.9 ± 3.1	42.7 ± 5.3	36.8 ± 2.6	**25.4** ± 3.0	33.0 ± 2.6	35.8 ± 2.0
Column_3C	61.8 ± 4.7	57.2 ± 3.7	70.3 ± 5.3	59.3 ± 4.0	59.8 ± 4.7	57.1 ± 2.4	**43.7** ± 2.1
Waveform1	1157 ± 11.9	1169 ± 25.2	1056 ± 24.9	1126 ± 12.8	**928** ± 27.4	1048 ± 15.7	1054 ± 11.5
Waveform2	1186 ± 24.7	1237 ± 23.1	1060 ± 32.3	1167 ± 15.2	**1008** ± 36.3	1069 ± 20.0	1060 ± 13.2
ALLAML	11.4 ± 8.0	13.6 ± 8.3	6.2 ± 7.5	11.6 ± 7.8	$*$	10.3 ± 2.1	**3.4** ± 1.0
Average	44.59 ± 3.81	43.23 ± 3.8	53.4 ± 5.71	43.83 ± 3.7	44.5 ± 3.4	38.81 ± 2.41	**31.91** ± **1.82**

Table 6 The number of TS and standard deviation for the FODT and the chosen benchmarks, and the best scores are indicated in boldface

Dataset	BFT	C4.5	LAD	SC	NBT	FRDT	FODT
Iris	9.3 ± 2.1	4.6 ± 0.6	7.0 ± 0.3	7.4 ± 2	4.4 ± 2.9	**4.0** ± 0.0	4.1 ± 0.1
Wine	10.6 ± 2.7	9.6 ± 1.2	13.0 ± 5.1	10.3 ± 3.2	**3.9** ± 2.6	4.2 ± 0.3	4.1 ± 0.1
Wdbc	16.5 ± 4.6	22.4 ± 3.9	16.2 ± 2.6	12.6 ± 4.4	18.2 ± 3.6	7.8 ± 0.4	**6.0** ± 0.7
Credit	30.3 ± 23.3	51.7 ± 12.1	8.8 ± 2.2	10.5 ± 10.6	14.2 ± 7.7	**7.3** ± 1.4	9.4 ± 1.8
Heart	28.8 ± 11.9	34.6 ± 5.7	15.6 ± 1.2	15.4 ± 8.1	9.6 ± 3.7	6.9 ± 1.1	**5.6** ± 1.3
Haberman	20.2 ± 22.5	21.8 ± 11.4	8.4 ± 1.9	**3.8** ± 3.8	9.9 ± 6.8	4.5 ± 0.5	4.0 ± 0.7
Newthyroid	13.6 ± 2.8	14.9 ± 2.1	11.8 ± 1.9	11.8 ± 3.6	7.6 ± 3.2	9.8 ± 1.0	**5.8** ± 0.4
Wobc	31.0 ± 12.4	23.5 ± 5.5	13.6 ± 3.2	15.9 ± 7.1	5.7 ± 5.6	10.0 ± 1.8	**5.3** ± 0.8
Column_3C	27.3 ± 11.2	23.2 ± 5.7	9.8 ± 0.9	13.3 ± 8.3	16.0 ± 5.1	7.7 ± 0.8	**7.3** ± 0.3
Waveform1	343.4 ± 89.2	541.5 ± 28.5	30.4 ± 1.8	125.7 ± 45.3	47.5 ± 36.9	**23.0** ± 2.0	26.4 ± 3.0
Waveform2	283.9 ± 97.7	591.9 ± 24.3	29.8 ± 2.6	98.3 ± 34.0	94.5 ± 43.4	4.1 ± 0.4	**4.0** ± 0.1
ALLAML	3.8 ± 1.0	4.3 ± 1.0	28.2 ± 1.8	**3.2** ± 0.6	*	8.7 ± 0.8	6.1 ± 0.2
Average	19.14 ± 9.45	21.06 ± 4.92	13.24 ± 2.11	10.42 ± 5.17	9.94 ± 4.58	7.09 ± 0.81	**5.77** ± **0.64**

It is worth noting that the NRS FS FAST is used flexibly. On the one hand, some datasets, due to their data distribution, may be empty after the reduction of attributes. At this moment we do not reduce the attributes. On the other hand, if the datasets are too large (such as waveform1 and waveform2), the efficiency of the NRS FS FAST itself will be low, it is not necessary to reduce the attributes at this time.

4.4.2 Comparison on time complexity

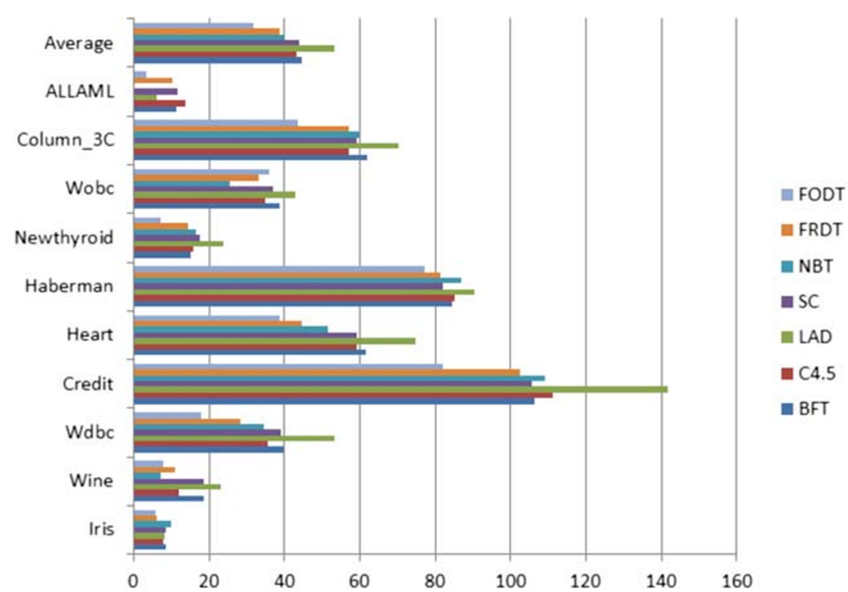
This subsection compares the time complexities of the FODT with its state-of-the-art competitors: C4.5 [11], BFT [39], LAD [40], SC [41], NBT [42] and FRDT [17]. The time complexities of the chosen benchmarks are shown in Table 7.

In order to compare the time complexity of our method with the state-of-the-art competitors more clearly, the datasets in Table 2 are used for complexity ranking. Without loss of generality, $logn > c$, therefore, the time complexities of BFT, LAD and SC are the same. When $MaxL = H$, the time complexities of FRDT and FODT are also the same. Thus, we only need to compare our method with BFT, C4.5, and NBT. The time complexity is ranked in ascending order, i.e. higher ranking means lower time complexity. Table 8 shows the time complexity ranking of BFT, C4.5, NBT and FODT. It can be observed that the average ranking of the FODT is better than BFT, C4.5 and NBT. Therefore, the proposed method has obvious advantages in terms of time complexity.

4.4.3 Holm test

To arrive at strong evidence, the statistical test is used to analyze whether the FODT is significantly better than other

Fig. 12 The number of MI for the FODT and the chosen benchmarks



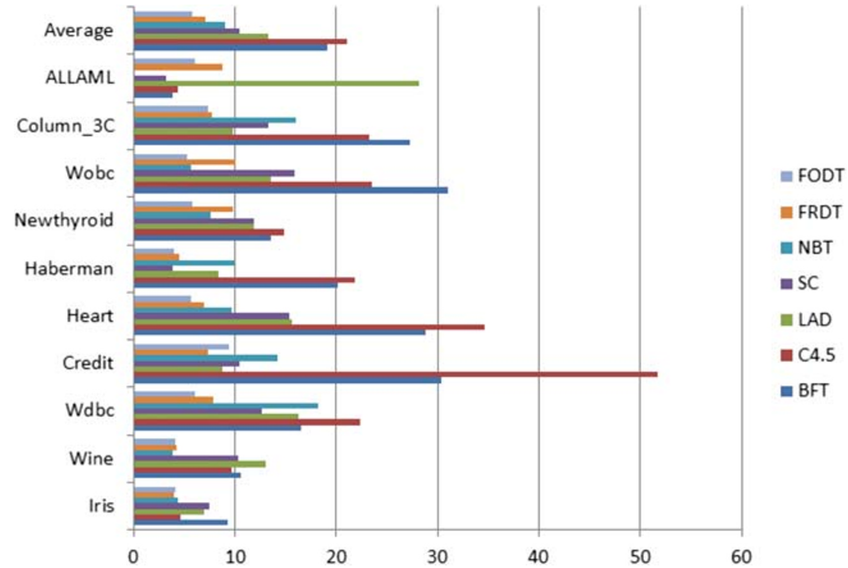


Fig. 13 The number of TS for the FODT and the chosen benchmarks

decision trees. Holm test [47] is applied in this paper, and the test statistic for comparing the *j*-th classifier and the *k*-th classifier is expressed as:

$$Z = \frac{Rank_j - Rank_k}{SE}, \tag{6}$$

$$SE = \sqrt{\frac{l(l+1)}{6 \times N}}, \tag{7}$$

$$Rank_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j, \tag{8}$$

where l is the number of classifiers, N is the number of datasets, r_i^j is the rank of the classifier j on the i-th dataset, and $Rank_j$ is the average rank of the classifier j on the entire dataset.

Statistic Z follows the standard normal distribution, and the z value is used to determine the corresponding probability p from the table of normal distribution. We denote the ordered p values by p_1, p_1, …, so that $p_1 \leq p_2 \leq \ldots \leq p_{l-1}$ and then compare p_j with $a/(1-j)$ (a stands for confidence level, usually is set to 0.05). If $p_1 < a/(l-1)$, the corresponding hypothesis (two classifiers have the same performance) should be rejected, and then we compare p_2 with $a/(l-2)$. If the second hypothesis is rejected, the test proceeds with the third one, etc. As long as there is a hypothesis cannot be rejected, all the remaining assumptions shall not be rejected.

According to Table 5 and Eq. (8), the average ranking of all decision trees can be obtained, $Rank_{LAD} = 5.75$, $Rank_{BFT} = 5.08$, $Rank_{SC} = 4.83$, $Rank_{C4.5} = 4.67$, $Rank_{NBT} = 3.45$, $Rank_{FRDT} = 2.33$, and $Rank_{FODT} = 1.58$. With $a = 0.05$, $l = 7$ and $N = 12$, the standard deviation $SE = 0.88$. The results of the Holm test are shown in Table 9, which indicate that the Holm procedure rejects the first four hypotheses since the corresponding p values are smaller than the adjusted $a's$, and only the last two hypotheses are accepted. This means that the classification accuracy of the FODT is significantly better than that of traditional decision trees NBT, SC, BFT, and LAD.

Table 7 The time complexity of the FODT and its state-of-the-art competitors

Algorithm	Time complexity
BFT	$O(m*n*logn)$
C4.5	$O(n^3)$
LAD	$O(m*n*logn+m*n*c)$
SC	$O(m*n*logn)$
NBT	$O(m^2*n*c*logn)$
FRDT	$O(MaxL*c*n*logn)$
FODT	$O(H*c*n*logn)$

Table 8 The time complexity ranking of BFT, C4.5, NBT and FODT on nine datasets

Dataset	Algorithm			
	BFT	C4.5	NBT	FODT
Iris	1	4	3	2
Wine	1	4	3	2
Wdbc	2	4	3	1
Credit	2	4	3	1
Heart	2	4	3	1
Haberman	1	4	3	2
Newthyroid	1	4	3	2
Wobc	1	4	3	2
Column_3C	1	4	3	2
Waveform1	2	4	3	1
Waveform2	2	4	3	1
ALLAML	3	2	4	1
Average	1.58	3.83	3.08	1.50

Table 9 The Holm test

No	Classifier	$\frac{Rank_j - Rank_{FODT}}{SE}$	Z	p	$\frac{a}{1-j}$
1	LAD	(5.75–1.58)/0.88	4.7386	5.3090e-06	0.0083
2	BFT	(5.08–1.58)/0.88	3.9773	1.4651e-04	0.01
3	SC	(4.83–1.58)/0.88	3.6932	4.3559e-04	0.0125
4	C4.5	(4.67–1.58)/0.88	3.5114	8.3849e-04	0.0167
5	NBT	(3.45–1.58)/0.88	2.1250	0.0417	0.025
6	FRDT	(2.33–1.58)/0.88	0.8523	0.2774	0.05

Although the FODT is not significantly higher than the C4.5 and FRDT, the average classification accuracy of the FODT is higher than that of the C4.5 and FRDT.

5 Conclusion

In this paper, we propose a novel fuzzy oblique decision tree, called FODT, which can achieve high classification accuracy and small tree size. Different from traditional axis-parallel decision trees and oblique decision trees, the FODT takes dynamic mining fuzzy rules as decision functions. In order to eliminate data redundancy and improve classification efficiency, the NRS_FS_FAST algorithm is first introduced to reduce attributes. Then, the FRGA is proposed to generate fuzzy rules, and these rules are used to construct leaf nodes for each class in each layer of the FODT. The growth of the FODT is developed by expanding an additional node, which is the only non-leaf node of each layer of the tree, and the fuzzy numbers of additional node in each layerare recalculated to get more accurate fuzzy partition. Finally, the parameter δ that can control the size of the tree is optimized by genetic algorithm. A series of comparative experiments on twenty UCI machine learning datasets and one biomedical dataset have verified the effectiveness of the proposed method. Therefore, our method is feasible and promising for dealing with the classification problem.

Acknowledgements This work was supported by the National Natural Science Foundation of China (61627809, 61433004, 61621004), and Liaoning Revitalization Talents Program (XLYC1801005).

References

1. Cover T, Hart P (2002) Nearest neighbor pattern classification[J]. IEEE Trans Inf Theory 13(1):21–27
2. Medin DL, Schaffer MM (2016) Context theory of classification learning[J]. Psychol Rev 85(3):207–238
3. Tsang ECC, Chen D, Yeung DS et al (2008) Attributes reduction using fuzzy rough sets[J]. IEEE Trans Fuzzy Syst 16(5):1130–1141
4. Jing Y, Li T, Fujita H et al (2017) An incremental attribute reduction approach based on knowledge granularity with a multi-granulation view[J]. Inf Sci 411:23–38
5. Zhang DW, Wang P, Qiu JQ et al (2010) An improved approach to feature selection[C]. In: Machine learning and cybernetics (ICMLC), 2010 international conference on, vol 1. IEEE, pp 488–493
6. Liu ZG, Pan Q, Dezert J (2014) A belief classification rule for imprecise data[J]. Appl Intell 40(2):214–228
7. Afify AA (2016) A fuzzy rule induction algorithm for discovering classification rules[J]. J Intell Fuzzy Syst 30(6):3067–3085
8. Park SB, Zhang BT, Kim YT (2003) Word sense disambiguation by learning decision trees from unlabeled data[J]. Appl Intell 19(1–2): 27–38
9. Li XB, Sweigart J, Teng J et al (2001) A dynamic programming based pruning method for decision trees[J]. INFORMS J Comput 13(4):332–344
10. Wu CC, Chen YL, Liu YH et al (2016) Decision tree induction with a constrained number of leaf nodes[J]. Appl Intell 45(3):673–685
11. Quinlan JR (2014) C4.5: programs for machine learning[M]. Elsevier, New York
12. Liang C, Zhang Y, Shi P et al (2015) Learning accurate very fast decision trees from uncertain data streams[J]. Int J Syst Sci 46(16): 3032–3050
13. Sok HK, Ooi MPL, Kuang YC et al (2016) Multivariate alternating decision trees[J]. Pattern Recogn 50:195–209
14. Kumar PSJ, Yung Y, Huan TL (2017) Neural network based decision trees using machine learning for Alzheimer's diagnosis[J]. Int J Comput Inform Sci 4(11):63–72
15. Shukla SK, Tiwari MK (2012) GA guided cluster based fuzzy decision tree for reactive ion etching modeling: a data mining approach[J]. IEEE Trans Semicond Manuf 25(1):45–56
16. Liu X, Feng X, Pedrycz W (2013) Extraction of fuzzy rules from fuzzy decision trees: an axiomatic fuzzy sets (AFS) approach[J]. Data Knowl Eng 84:1–25
17. Wang X, Liu X, Pedrycz W et al (2015) Fuzzy rule based decision trees[J]. Pattern Recogn 48(1):50–59
18. Segatori A, Marcelloni F, Pedrycz W (2018) On distributed fuzzy decision trees for big data[J]. IEEE Trans Fuzzy Syst 26(1):174–192
19. Tan PJ, Dowe DL (2006) Decision forests with oblique decision trees[C]. In: Mexican international conference on artificial intelligence. Springer, Berlin, pp 593–603
20. Barros RC, Jaskowiak PA, Cerri R et al (2014) A framework for bottom-up induction of oblique decision trees[J]. Neurocomputing 135:3–12
21. Do TN, Lenca P, Lallich S (2015) Classifying many-class high-dimensional fingerprint datasets using random forest of oblique decision trees[J]. Vietnam journal of computer science 2(1):3–12
22. Wickramarachchi DC, Robertson BL, Reale M et al (2016) HHCART: An oblique decision tree[J]. Comput Stat Data Anal 96:12–23
23. Rivera-Lopez R, Canul-Reich J, Gómez JA et al (2017) OC1-DE: a differential evolution based approach for inducing oblique decision trees[C]. In: International conference on artificial intelligence and soft computing. Springer, Cham, pp 427–438
24. Rivera-Lopez R, Canul-Reich J (2017) A global search approach for inducing oblique decision trees using differential evolution[C]. In: Canadian conference on artificial intelligence. Springer, Cham, pp 27–38
25. Narayanan SJ, Paramasivam I, Bhatt RB et al (2015) A study on the approximation of clustered data to parameterized family of fuzzy membership functions for the induction of fuzzy decision trees[J]. Cybernetics and Information Technologies 15(2):75–96
26. Han NM, Hao NC (2016) An algorithm to building a fuzzy decision tree for data classification problem based on the fuzziness intervals matching[J]. Journal of Computer Science and Cybernetics 32(4): 367–380

27. Sardari S, Eftekhari M, Afsari F (2017) Hesitant fuzzy decision tree approach for highly imbalanced data classification[J]. Appl Soft Comput 61:727–741
28. Ludwig SA, Picek S, Jakobovic D (2018) Classification of Cancer data: analyzing gene expression data using a fuzzy decision tree algorithm[M]. In: Operations research applications in health care management. Springer, Cham, pp 327–347
29. Manwani N, Sastry PS (2012) Geometric decision tree[J]. IEEE Trans Syst Man Cybern B Cybern 42(1):181–192
30. Patil SP, Badhe SV (2015) Geometric approach for induction of oblique decision tree[J]. International Journal of Computer Science and Information Technologies 5(1):197–201
31. Cant-Paz E, Kamath C (2000) Using evolutionary algorithms to induce oblique decision trees[C]. In: Proceedings of the 2nd annual conference on genetic and evolutionary computation. Morgan Kaufmann Publishers Inc, pp 1053–1060
32. Skowron A (2000) Rough sets in KDD[J]. Special invited speaking, WCC pp 1–17
33. Zhang-Yan XU, Hou W, Song W et al (2009) Efficient heuristic attribute reduction algorithm based on information entropy[J]. Journal of Chinese Computer Systems 30(9):1805–1810
34. Yang SM, Jie M, Zhang ZB et al (2015) An improved heuristic attribute reduction algorithm based on information entropy in rough set[J]. Open Cybernetics & Systemics Journal 9(1):2774–2779
35. Guang-Yuan FU, Han-Zhao WU, Yang XG (2013) Attribute reduction algorithm of rough set based on the combined compatibility and importance of attributes[J]. Science Technology & Engineering
36. Zadeh LA (1997) Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic[J]. Fuzzy Sets Syst 90(2):111–127
37. Agrawal R, Imielinski T, Swami A (1993) Database mining: a performance perspective[J]. IEEE Trans Knowl Data Eng 5(6):914–925
38. Wang X, Liu X, Pedrycz W et al (2012) Mining axiomatic fuzzy set association rules for classification problems[J]. Eur J Oper Res 218(1):202–210
39. Shi H (2007) Best-first decision tree learning[D]. The University of Waikato
40. Holmes G, Pfahringer B, Kirkby R et al (2002) Multiclass alternating decision trees[C]. In: European conference on machine learning. Springer, Berlin, pp 161–172
41. Breiman LI, Friedman JH, Olshen RA et al (1984) Classification and regression trees (CART)[J]. Encyclopedia of Ecology 40(3): 582–588
42. Kohavi R (1996) Scaling up the accuracy of naive-Bayes classifiers: a decision-tree hybrid[C]. In: KDD 96, pp 202–207
43. Mirzamomen Z, Kangavari MR (2017) A framework to induce more stable decision trees for pattern classification[J]. Pattern Anal Applic 20(4):991–1004
44. Asuncion A, Newman D (2007) UCI machine learning repository[J]. http://www.ics.uci.edu/mlearn/MLRepository.html. Accessed 26 Jan 2018
45. http://stanford.edu/marinka/nimfa/nimfa.datasets.html. Accessed 28 July 2018
46. Witten LH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, third edition[J]. ACM SIGMOD Rec 31(1):76–77
47. Hhn JC, Hllermeier E (2009) FR3: a fuzzy rule learner for inducing reliable classifiers[J]. IEEE Trans Fuzzy Syst 17(1):138–149

Yuliang Cai received the B.S. degree in information and computing science from the Ludong University, Yantai, China in 2014, and the M.S. degree in control theory and control engineering from the Dalian University of Technology, Dalian, China in 2017. She is currently working toward the Ph.D. degree in control theory and control engineering at Northeastern University, Shenyang, China. Her research interests include multi-agent system control, fuzzy control, adaptive dynamic programming, etc.

Qiang He received the B.S. degree in information and computing science from the Ludong University, Yantai, China in 2014, and the M.S. degree in communication and information systems from the Northeastern University, Shenyang, China in 2016. He is currently working toward the Ph.D. degree in computer application technology at Northeastern University, Shenyang, China. His research interests include social network analytic, data mining, software defined networking, etc.

Huaguang Zhang received the B.S. degree and the M.S. degree in control engineering from Northeast Dianli University of China, Jilin City, China, in 1982 and 1985, respectively. He received the Ph.D. degree in thermal power engineering and automation from Southeast University, Nanjing, China, in 1991. He joined the Department of Automatic Control, Northeastern University, Shenyang, China, in 1992, as a Postdoctoral Fellow for two years. Since 1994, he has been a Professor and Head of the Institute of Electric Automation, School of Information Science and Engineering, Northeastern University, Shenyang, China. His main research interests are fuzzy control, stochastic system control, neural networks based control, nonlinear control, and their applications. He has authored and coauthored over 280 journal and conference papers, six monographs and co-invented 90 patents. Dr. Zhang is the fellow of IEEE, the E-letter Chair of IEEE CIS Society, the former Chair of the Adaptive Dynamic Programming & Reinforcement Learning Technical Committee on IEEE Computational Intelligence Society. He is an Associate Editor of AUTOMATICA, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON CYBERNETICS, and NEUROCOMPUTING, respectively. He was an Associate Editor of IEEE TRANSACTIONS ON FUZZY SYSTEMS (2008-2013). He was awarded the Outstanding Youth Science Foundation Award from the National Natural Science Foundation Committee of China in 2003. He was named the Cheung Kong Scholar by the Education Ministry of China in 2005. He is a recipient of the IEEE Transactions on Neural Networks 2012 Outstanding Paper Award.

Jie Duan received the B.S. degree from Northeast Electric Power University, Jilin, China, in 2013, and the M.S. degree from Northeast Electric Power University, Jilin, China, in 2016. She is pursing the Ph.D. degree in Northeastern University, Shenyang, China. Her research interests include multi-agent systems, switching systems, adaptive control, finite time control.