

## A Study on the Approximation of Clustered Data to Parameterized Family of Fuzzy Membership Functions for the Induction of Fuzzy Decision Trees

Swathi J. Narayanan<sup>1</sup>, Ilango Paramasivam<sup>1</sup>, Rajen B. Bhatt<sup>2</sup>, M. Khalid<sup>3</sup>

<sup>1</sup>SCSE, VIT University, Vellore-632014, TamilNadu, India

<sup>2</sup>Robert Bosch Engineering and Business Solutions Ltd., Bangalore, India

<sup>3</sup>Galgotias University, Greater Noida, India

Emails: jnswathi@vit.ac.in      pilango@vit.ac.in      rajen.bhatt@gmail.com  
mkhalidbs@yahoo.com

**Abstract:** This paper investigates the Triangular, Trapezoidal and Gaussian approximation methods for the purpose of induction of Fuzzy Decision Trees (FDT). The generation of FDT is done using a Fuzzy ID3 induction algorithm. In this work three fuzzy partitioning techniques which form the basis for our investigation are given attention, namely Fuzzy C Means clustering (FCM), Grid partitioning and Subtractive clustering (Subclust). Our contribution lies in studying the effect of various approximations on the generation of FDT giving specific attention to the classification accuracy of FDT. In this study we show that the accuracy levels of FDT generated using FCM clustered raw data, bypassing the approximation step, is acceptable and this method has several advantages too. Several computational experiments are conducted and non parametric statistical tests are performed to find if any significant differences exist between the method of bypassing the approximation step and the other methods which include approximation. Ten FDTs are developed and used in this study. These FDT's differ in their fuzzy partitioning techniques and the approximation methods used.

**Keywords:** Fuzzy decision tree, clustering, fuzzy membership functions, non parametric statistical test.

## 1. Introduction

The main motivation of any classification study is to design an accurate classifier by taking advantage of the past experience stored in the form of a large amount of raw data and to uncover the predictive structure of the problem for extracting human-interpretable domain knowledge. Out of many classification techniques, the decision tree classifier is commonly used to obtain human interpretable domain knowledge due to its capability to break down a complex decision making method into a collection of locally optimal simple decisions through top-down greedy search techniques [1]. Although the decision trees generated, using an ID3 family and C4.5 are useful in building knowledge-based expert systems, they are often not capable of handling cognitive uncertainties consistent with human information processing, such as vagueness and ambiguity. In general, vagueness is related to the difficulty in making sharp classification boundaries and ambiguity is related to one-to-many mappings observed in the patterns. To overcome these deficiencies, by incorporating the fuzzy uncertainty measure in a crisp decision tree generation algorithm like ID3, various researchers have developed FDT induction algorithms [2-6].

There are several applications where FDT is used for extracting human interpretable classification rules [7-10]. The most important task in the induction of FDT is fuzzy clustering of the input data and its approximation to various membership functions. In the development of fuzzy systems, one frequently faces the problem of adequate choice of the Membership Functions (MF). The good choice of membership functions is crucial to the behaviour of fuzzy systems. In many cases there are no theoretical criteria that would justify the use of one or another function, and they are selected based on their suitability to empirical data. There is a large amount of fuzzy data in almost every real life field. For conversion of any clustered data into trapezoidal membership functions, a convex hull method is proposed by N a k a n i s h i et. al. [11]. However, the methodology is not very clear and the implementation depends very much on the case-by-case basis. In addition, the methodology generates membership functions that are either highly overlapped or highly separated. None of the papers [11, 12] mention the trapezoidal approximation algorithm clearly. The triangular membership functions are another important class of membership functions to model various linguistic modelling scenarios that have not been adequately addressed. To overcome these limitations we introduced two heuristic approximation algorithms [13].

In this paper we aim to study the effect of approximation of clustered raw data to various membership functions for the purpose of induction of FDT. We claim that the approximation of any clustered raw data to various membership functions involves additional processing before beginning of the tree induction process. Also, it requires additional parameters to be stored in the main memory, i.e., the parameters  $\{a, b, c\}$  for triangular, the parameters  $\{a, b, c, d\}$  for trapezoidal, and the parameters  $\{\text{center}, \text{sigma}\}$  for Gauss; while a simple fuzzy cluster can be represented by just one parameter, representing the cluster center. In many cases, the approximation may also require experts' advice to decide the shape of the

membership functions and different attributes may need different types of membership functions or even a combination of membership functions to perfectly model the vagueness. The paper is organized as follows: In Section 2 a FDT induction process is presented along with clustering techniques and approximation methods. The computational experimental results are reported in Section 3. Inferential statistics of the results obtained is discussed in Section 4 and finally, concluding remarks are given in Section 5.

## 2. Induction of a fuzzy decision tree

In this section the fuzzy clustering techniques and the approximation methods which form the basis for our investigation are discussed along with FDT induction process.

### 2.1. Fuzzy clustering

Fuzzy clustering is a process of assigning fuzzy partition labels and the associated degrees of membership to each pattern in  $X$ . Let  $X = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n\}$  be a set of  $n$  feature vectors and each feature vector  $\mathbf{x}^i$  be a point in a  $p$ -dimensional observation (or feature) space. In other words, each pattern  $\mathbf{x}^i$  has been described by a set of  $p$  attributes  $\{x_1, x_2, \dots, x_p\}$ . Each attribute  $x_j$  measures some important feature (or property, condition) of the system under study.

#### 2.1.1. Fuzzy C means clustering

The general Fuzzy C Means (FCM) algorithm [14] generates  $c$ -partitions of  $X$  in the form of a  $c \times n$  matrix, whereas the FCM considered for our study generates  $c_j$  partitions for the values of the attribute  $x_j$  in the form of a  $c \times n \times j$  matrix, where for each attribute  $x_j$ , the partition matrix is  $U = \mu_{F_{jk}}(x_j^i)$ . FCM allows gradual cluster membership, i.e., in effect, the data can belong to more than one cluster at the same time. The family of the partition matrix  $U$  considered for our experimental study is

$$(1) \quad M_{fcj} = \left\{ \mathbf{U} \in R^{cnj} \mid \sum_{k=1}^{c_j} \mu_{F_{jk}}(x_j^i) = 1 \quad \forall i \right\},$$

where  $\mu_{F_{jk}}(x_j^i)$  is the membership degree of the  $i$ -th pattern in  $k$ -th fuzzy cluster of the attribute  $x_j$ .

#### 2.1.2. Grid partitioning

Grid Partitioning generates membership values by partitioning the input space into a given number of intervals whose limits do not necessarily have any physical meaning [15].

### 2.1.3. Subtractive clustering

The subtractive clustering method assumes that each data point is a potential cluster center and calculates a measure of the likelihood that each data point would define as the cluster center, based on the density of the surrounding data points [16].

## 2.2. Approximation methods for FCM and subtractive clustered data

Two efficient heuristic algorithms proposed in our earlier work [13] are used for converting FCM clustered raw data and subtractive clustered Gaussian data to parameterized family of fuzzy membership functions, like Triangular MF and Trapezoidal MF.

### 2.2.1. Triangular approximation

A fuzzy set  $F$  with a triangular membership function is specified by three parameters  $\{a, b, c\}$  where the degree of membership of pattern  $u$  to the triangular fuzzy set  $F$  is

$$(2) \quad \mu_F(u) = \begin{cases} 0, & u \leq a, \\ \frac{u-a}{b-a}, & a < u \leq b, \\ \frac{u-b}{c-b}, & b < u < c, \\ 0, & u \geq c. \end{cases}$$

Let  $\{a, b, c\}$ , where  $a < b < c$ , determines the three vertices of the triangular membership function on  $X$ . A continuous attribute  $x_j$  is fuzzified into  $c_j$  triangular fuzzy sets.

The pseudo code of the proposed algorithm for obtaining the parameters  $\{a, b, c\}$  for triangular membership functions from FCM/Subtractive cluster matrix  $U$  and a center vector (or matrix)  $C$  is given in Algorithm 1.

The symbols used in the pseudo code [13] are:

$x_{j\_sort}$  is the ascending order of sorting the patterns of  $x_j$ ,

$x_{j\_index}$  are the indices of the sorted patterns,

$U$  is the cluster matrix generated using FCM,

$U_k$  is the  $k$ -th row of matrix  $U$ ,

$U_{k\_sort}$  is the arrangement of  $U_k$  with respect to  $x_{j\_index}$ ,

$C$  is the center vector (or matrix),

$a_{index}$  is the index of the pattern of  $x_j$  corresponding to parameter  $a$ , and  $c_{index}$  is the index of the pattern of  $x_j$  corresponding to the parameter  $c$ .

**Algorithm 1** [13]. Triangular approximation algorithm

*Input:*  $U, C$

*Output:*  $\{a, b, c\}$  for a Triangular membership function  $F_k$  ( $k = 1, \dots, c_j$ )

**Start**

calculate  $x_{j\_sort}$

```

calculate  $x_{j\_index}$ 
for  $k=1:c_j$ 
  calculate  $U_{k\_sort}$ 
   $\min U = \text{minimum}(U_{k\_sort}) + 0.05$ 
   $\text{ucut} = \text{min\_max indices of } U_{k\_sort} \text{ equal to max } U$ 
   $L = x_{j\_sort}(\text{ucut}(1))$ 
   $R = x_{j\_sort}(\text{ucut}(2))$ 
   $b$  is the average of  $L$  and  $R$ 
   $\text{lcut} = \text{indices of } U_{k\_sort} \text{ less than } \min U$ 
   $a_{index} = \text{an element from lcut which is less than } b \text{ and maximum of lcut}$ 
   $c_{index} = \text{an element from lcut which is greater than } b \text{ and minimum of lcut}$ 
   $a = x_{j\_sort}(a_{index})$ 
   $c = x_{j\_sort}(c_{index})$ 
  if ( $a = \text{'null'}$ )
    if  $U_{k\_sort}(1) = 1$  or  $U_{k\_sort}(1) = 0$ 
       $a = x_{j\_sort}(1)$ 
    else
       $a = x_{j\_sort}(1) - (U_{k\_sort}(1) / (1 - U_{k\_sort}(1))) * (b - x_{j\_sort}(1))$ 
    end
  end
  if ( $c = \text{'null'}$ )
    if  $U_{k\_sort}(n) = 1$  or  $U_{k\_sort}(n) = 0$ 
       $c = x_{j\_sort}(n)$ 
    else
       $c = x_{j\_sort}(n) + (U_{k\_sort}(n) / (1 - U_{k\_sort}(n))) * (b - a)$ 
    end
  end
end
end
Stop

```

### 2.2.2. Trapezoidal approximation

A fuzzy set  $F$  with a trapezoidal membership function is specified by a set of four parameters  $\{a, b, c, d\}$  where the degree of membership of pattern  $u$  to the trapezoidal fuzzy set  $F$  is given by

$$(3) \quad \nu_F(u) = \begin{cases} 0, & u \leq a, \\ \frac{u-a}{b-a}, & a < u < b, \\ 1, & b \leq u \leq c, \\ \frac{u-c}{d-c}, & c < u < d, \\ 0, & u \geq d. \end{cases}$$

The pseudo code of the proposed algorithm for obtaining the trapezoidal parameters  $\{a, b, c, d\}$  is given in Algorithm 2. The symbols used in the pseudo

code are the same as explained in Section 2.2.1.

**Algorithm 2 [13].** Trapezoidal approximation algorithm

*Input:*  $U, C$

*Output:*  $\{a, b, c, d\}$  for a Trapezoidal membership function  $F_k$  ( $k = 1, \dots, c_j$ )

**Start**

calculate  $x_{j\_sort}$

calculate  $x_{j\_index}$

**for**  $k=1:c_j$

    calculate  $U_{k\_sort}$

$\min U = \text{minimum}(U_{k\_sort}) + 0.05$

$\max U = \text{maximum}(U_{k\_sort}) - 0.05$

$ucut = \text{min\_max indices of } U_{k\_sort} \text{ greater than } \max U$

$lcut = \text{indices of } U_{k\_sort} \text{ less than } \min U$

$b = x_{j\_sort}(ucut(1))$

$c = x_{j\_sort}(ucut(2))$

$a_{index} = \text{an element from } lcut \text{ which is less than } ucut(1) \text{ and maximum of } lcut$

$d_{index} = \text{an element from } lcut \text{ which is greater than } ucut(2) \text{ and minimum of } lcut$

$a = x_{j\_sort}(a_{index})$

$d = x_{j\_sort}(d_{index})$

**if** ( $a = 'null'$ )

**if**  $U_{k\_sort}(1) = 1$  or  $U_{k\_sort}(1) = 0$

$a = x_{j\_sort}(1)$

**else**

$a = x_{j\_sort}(1) - (U_{k\_sort}(1) / (1 - U_{k\_sort}(1))) * (b - x_{j\_sort}(1))$

**end**

**end**

**if** ( $d = 'null'$ )

**if**  $U_{k\_sort}(n) = 1$  or  $U_{k\_sort}(n) = 0$

$d = x_{j\_sort}(n)$

**else**

$d = x_{j\_sort}(n) + (U_{k\_sort}(n) / (1 - U_{k\_sort}(n))) * (c - x_{j\_sort}(1))$

**end**

**end**

**end**

**Stop**

### 2.2.3. Gaussian approximation

The center of each Gaussian membership function has been initialized by cluster centers generated by the FCM algorithm. To initialize the standard deviations (sigma) we have used a value, proportional to the minimum distance between centers [17]. The pseudo code to obtain Gaussian parameters from FCM clustered centers is given in Algorithm 3.

The symbols used in the pseudo code are:

$V$  is the Cluster matrix,

$C$  is the Cluster center vector,

$\lambda$  is the dispersion parameter,  $0 < \lambda \leq 1$ .

**Algorithm 3.** Gaussian approximation algorithm

*Input:*  $C, \lambda$

*Output:* Standard deviation  $\{S\}$  for  $F_{jk}$  ( $k = 1, \dots, c_j$ )

**for** each variable  $x_j$  ( $j = 1, \dots, p$ )

**for** each membership function  $F_{jk}$  ( $k = 1, \dots, c_j$ )

Calculate the Euclidean distance  $dc(c_{jk}, c_{jh}), h = 1, \dots, c_j, h \neq k$

Calculate  $\sigma_{jk} = \lambda * dc_{\min}(c_{jk}, c_{jh}); 0 < \lambda \leq 1$

**end**

**end**

### 2.3. Approximations using Grid partitioning technique

The fuzzy set  $F$  is determined using triangular, trapezoidal, and Gaussian membership functions. The parameters for these MFs are generated using Grid partitioning technique available in MATLAB [18]. The pseudo code given in Algorithm 4 and Algorithm 5 describes the approach for generating parameters for various membership functions using grid partitioning technique.

The symbols used in the pseudo code are:

$X$ : Data matrix

$c$ : number of clusters

mftype: type of the membership function.

**Algorithm 4.** Triangular and trapezoidal membership function parameters estimation using Grid partitioning technique

*Input:*  $X, c, \text{mftype}$

*Output:*  $\{a, b, c\}$  for a Triangular MF and  $\{a, b, c, d\}$  for a Trapezoidal MF,  $F_k$  ( $k = 1, \dots, c_j$ )

**Start**

**for** each of the numerical attribute  $j$  in  $X$

mf\_n(j)= Initialize the number of mf's for  $j$

range(j, 1)= minimum of  $(x_j)$

range(j, 2)= maximum of  $(x_j)$

avalue =(range(j,2) – range(j, 1))/2/(mf\_n(j) – 1)

$b(1:\text{mf}_n(j))=\text{linspace}(\text{range}(j, 1), \text{range}(j, 2), \text{mf}_n(j))'$

**if** mftype is 'trimf'

**for**  $i = 1:\text{mf}_n(j)$

$a(i) = b(i) - 2*\text{avalue}$

$c(i) = b(i) + 2*\text{avalue}$

**end**

**else**

```

for  $i = 1:mf\_n(j)$ 
     $a(i) = bvalue(i) - 1.4 * avalue$ 
     $b(i) = bvalue(i) - 0.6 * avalue$ 
     $c(i) = bvalue(i) + 0.6 * avalue$ 
     $d(i) = bvalue(i) + 1.4 * avalue$ 
end
end
end

```

**Algorithm 5.** Gaussian membership function parameters estimation using Grid partitioning technique

*Input:*  $X, c$

*Output:*  $\{S, C\}$  for Gaussian MF  $F_k (k = 1, \dots, c_j)$

**Start**

**for** each of the numerical attribute  $j$  in  $X$

$mf\_n(j) =$  Initialize the number of mfs for  $j$

$range(j, 1) =$  minimum of  $(x_j)$

$range(j, 2) =$  maximum of  $(x_j)$

$sigma = [ (range(j, 2) - range(j, 1)) / 2 / (mf\_n(j) - 1)] / \sqrt{2 * \log(2)}$

$S = sigma * ones(mf\_n(j), 1)$

$C(1:mf\_n(j)) = \text{linspace}(range(j, 1), range(j, 2), mf\_n(j))'$

**end**

**end**

The “linspace” function [18] in Algorithms 4 and 5 determines the center of the generated membership functions in such a way that they are always equally spaced along the domain of an input variable  $x_j$ , where the domain is determined as the interval between the minimum and maximum of the corresponding  $x_j$ . Using the center  $\{b\}$ , the left and right cut points  $\{a, c\}$  are determined for each triangular membership function. The  $\{a, b, c, d\}$  parameters determine the four vertices of the trapezoidal membership function. The center point of each membership function is used to calculate the left cut point  $\{a\}$ , the height of the membership function  $\{b, c\}$  and the right cut point  $\{d\}$ . The pseudocode in Algorithm *d* determines the sigma (standard deviation  $\{S\}$ ) and center  $\{C\}$  for the Gaussian membership function.

#### 2.4. Fuzzy ID3

One popular approach for the induction of FDTs is the Fuzzy ID3 [2] algorithm, which is a fuzzy version of the ID3 algorithm, initially developed by Quinlan [19] for the crisp partitions. Fuzzy ID3 utilizes fuzzy classification entropy of a probability distribution to determine the best node during decision tree generation. For each fuzzy set  $\{F_{jk} \mid j = 1, \dots, p; k = 1, \dots, c_j\}$  of the attribute  $x_j$ , the certainty factor concerning the  $l$ -th class is defined as



$$(4) \quad \beta_{jk}^l = \frac{\sum_{i=1}^n \min\{\mu_{F_{jk}}(x_j^i), \mu_l(y^i)\}}{\sum_{i=1}^n \mu_{F_{jk}}(x_j^i)}, \quad 0 \leq \beta_{jk}^l \leq 1.$$

The fuzzy classification entropy of  $F_{jk}$  is defined as

$$(5) \quad \text{Entr}_{jk} = -\sum_{l=1}^q \beta_{jk}^l \times \log_2(\beta_{jk}^l).$$

The averaged fuzzy classification entropy of  $x_j$  is defined as

$$(6) \quad E_j = \sum_{k=1}^{c_j} w_{jk} \times \text{Entr}_{jk},$$

where  $w_{jk}$  denotes the weight of the  $k$ -th fuzzy set of the  $j$ -th attribute and is defined as

$$(7) \quad w_{jk} = \frac{\sum_{i=1}^n \mu_{F_{jk}}(x_j^i)}{\sum_{k=1}^{c_j} \left( \sum_{i=1}^n \mu_{F_{jk}}(x_j^i) \right)}.$$

The general procedure for generating FDT using Fuzzy ID3 is outlined as follows:

**Prerequisites:** Fuzzy partition space, leaf selection threshold  $\beta_{th}$ , best node selection criterion.

**Procedure:**

While there exist candidate nodes DO

Select one of them using a search strategy,

Generate its child-nodes,

Child-nodes, meeting the leaf threshold have to be levelled as leaf-nodes, otherwise the remaining child-nodes are regarded as new candidate nodes and the procedure is repeated until the stopping criterion is met.

### 3. Computational experiments

To experiment with the fuzzy partitioning techniques and various approximation methods described in Section 2, totally 10 FDTs are developed in three experimental setups. In the first setup, the fuzzy partitioning of each individual attribute in the input space  $X$  is performed, using FCM clustering technique. The FCM clustered raw data (FCM-Noapp) is approximated to triangular membership functions (FCM-Tri), trapezoidal membership functions (FCM-Trap) and Gaussian membership functions (FCM-Gauss). Here four FDTs are developed using Fuzzy ID3, taking Triangular approximated FCM values, Trapezoidal approximated FCM values, Gaussian approximated FCM values and finally taking FCM generated membership values directly without performing any approximation.

In the second setup, a similar kind of an experiment is conducted by partitioning the inputs using the *genparam* function available in MATLAB [18] for grid partitioning technique. In this case three FDTs are developed using Fuzzy ID3 taking Triangular approximated grid partitioned membership values (GRID-Tri), Trapezoidal approximated grid partitioned membership values (GRID-Trap) and Gaussian approximated grid partitioned membership values (GRID-Gauss).

In the third setup, for subtractive clustering, a *subclust (radii, data)* function available in MATLAB [18] is used to obtain the parameters of a Gaussian function. The *radii* parameter controls the number of clusters to be formed. If a small value is chosen for radii, the number of clusters formed would be large, in the same way if *radii* is increased to a bigger value, few numbers of clusters would be formed. Depending on the dataset properties, the choice of radii would differ. For all the experiments conducted, we started with a small random value and increased/decreased the radii in such a way that it leads to three numbers of clusters. The three numbers of clusters are chosen only for experimental purposes. Actually one has to obtain the optimal number of clusters for each numerical attribute in the input space. Here we do not consider the optimal cluster number estimation within the scope of this paper. In this setup, three more FDTs are developed using subtractive clustered membership values (Subclust-Gauss) and its approximation to triangular (Subclust-tri) and trapezoidal (Subclust-trap) ones.

Table 1. Sports dataset

Outlook	Temp (f)	Humidity (%)	Windy?	Decision
Rain	71	96	True	<b>Don't play</b>
Rain	65	70	True	<b>Don't play</b>
Overcast	72	90	True	<b>Play</b>
Overcast	83	78	False	<b>Play</b>
Overcast	83	78	False	<b>Play</b>
Rain	75	80	False	<b>Play</b>
Overcast	64	65	True	<b>Play</b>
Sunny	75	70	True	<b>Play</b>
Sunny	80	90	True	<b>Don't play</b>
Sunny	85	85	False	<b>Don't play</b>
Overcast	81	75	False	<b>Play</b>
Rain	68	80	False	<b>Play</b>
Rain	70	96	False	<b>Don't play</b>
Sunny	72	95	False	<b>Don't play</b>
Sunny	69	70	False	<b>Play</b>

Here we have considered the Sports dataset [20] given in Table 1 to demonstrate the results of the approximation methods of all fuzzy partitioning techniques considered. Out of the four attributes in the sports dataset, *Temp* and *Humidity* are numerical attributes. They are fuzzified into three fuzzy clusters using the FCM algorithm. *Windy* and *Outlook* are categorical attributes fuzzified into fuzzy singletons. Each pattern is classified to either **Play** or **Don't Play**.

### 3.1. Triangular approximations

The FCM clustered raw data and subtractive clustered Gaussian data are approximated to a triangular membership function using Algorithm 1, given in Section 2. The grid partitioning technique and its triangular approximations are obtained using Algorithm 4.

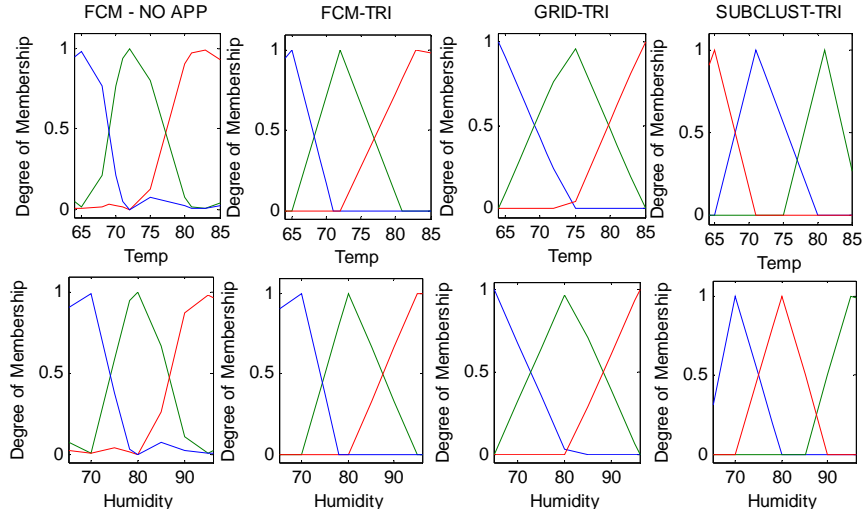


Fig. 1. Triangular approximations

Table 2. Triangular membership function parameters

Clustering technique	TRI								
	First fuzzy partition			Second fuzzy partition			Third fuzzy partition		
	$a$	$b$	$c$	$a$	$b$	$c$	$a$	$b$	$c$
FCM-Temp	46	65	71	65	72	81	72	83	246.22
Grid-Temp	53.5	64	74.5	64	74.5	85	74.5	85	95.5
Subtractive-Temp	54.56	65	71	65	71	80	75	81	86.49
FCM-Humidity	15.76	70	78	70	80	95	80	95	585.23
Grid-Humidity	49.5	65	80.5	65	80.5	96	80.5	96	111.5
Subtractive-Humidity	62.70	70	80	70	80	90	85	95	307.26

The graphs of FCM degree of the membership (FCM-Noapp) of each pattern along with the value of *Temp* and *Humidity* attribute are shown in Fig. 1 along with its approximations to Triangular Membership Functions (Triangular MF) using different clustering techniques. The Triangular MF parameters obtained from Fig 1. are given in Table 2.

### 3.2. Trapezoidal approximations

A similar experiment is conducted to obtain Trapezoidal approximations of FCM clustered raw data and subtractive clustered raw data using Algorithm 2. The grid partitioning and its trapezoidal approximations are obtained using Algorithm 4.

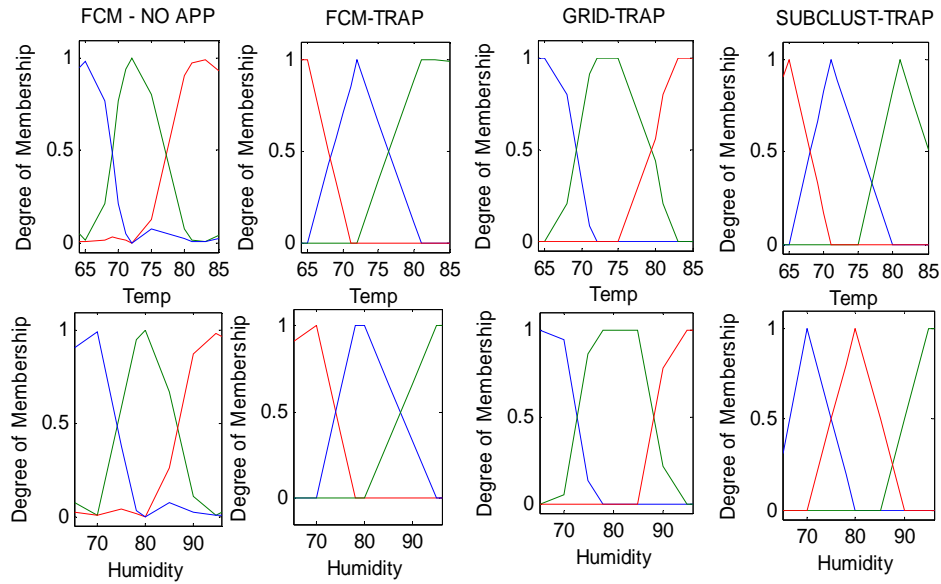


Fig. 2. Trapezoidal approximations

Table 3. Trapezoidal membership function parameters

Clustering technique	TRAP											
	First fuzzy partition				Second fuzzy partition				Third fuzzy partition			
	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
FCM-Temp	64	64	65	71	65	72	72	81	72	81	83	363.47
Grid-Temp	56.65	60.85	67.15	71.35	67.15	71.35	77.65	81.85	77.65	81.85	88.15	92.35
Subtractive -Temp	54.56	65	65	71	65	71	71	80	75	81	81	89.23
FCM-Humidity	15.76	70	70	78	70	78	80	95	80	95	96	1106.8
Grid-Humidity	54.15	60.35	69.65	75.85	69.65	75.85	85.15	91.35	85.15	91.35	100.65	106.85
Subtractive-Humidity	62.70	70	70	80	70	80	80	90	85	95	96	750.91

The graphs of the degrees of membership of each pattern approximated to the Trapezoidal MF are shown in Fig. 2. The Trapezoidal MF parameters obtained from Fig. 2 are given in Table 3.

### 3.3. Gaussian approximations

The FCM clustered raw data is approximated to Gaussian MF using Algorithm 3. The Grid partitioning and its Gaussian approximations are obtained using Algorithm 5. The graphs of the Gaussian degree of membership of each pattern obtained using different clustering techniques are given in Fig. 3. The parameters of the graphs generated in Fig. 3 are given in Table 4.

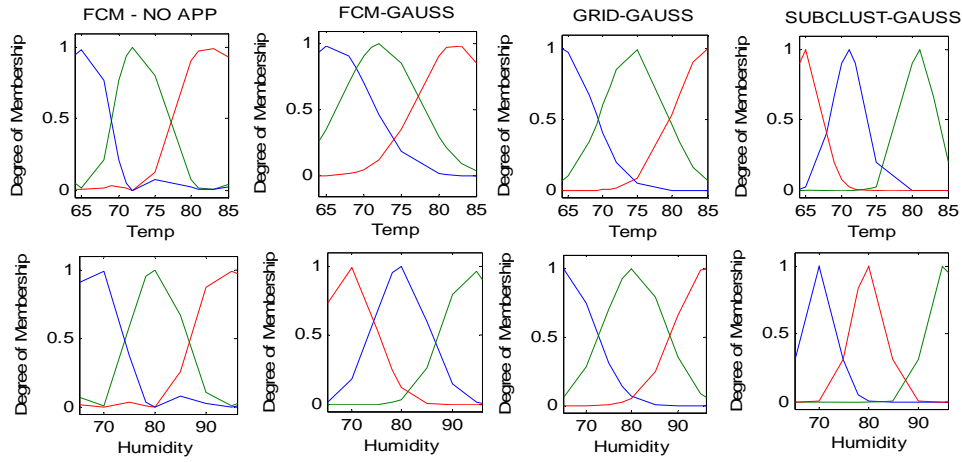


Fig. 3. Gaussian approximations

Table 4. Gaussian membership function parameters

Clustering technique	GAUSS					
	First fuzzy partition		Second fuzzy partition		Third fuzzy partition	
	Center $C$	Sigma $S$	Center $C$	Sigma $S$	Center $C$	Sigma $S$
FCM-Temp	65.7882	5.0122	72.1920	5.0122	82.2164	5.0122
Grid-Temp	64	4.4589	74	4.4589	85	4.4589
Subtractive-Temp	65	2.2274	71	2.2274	81	2.2274
FCM-Humidity	69.1630	5.2640	79.6909	5.2640	93.5592	5.2640
Grid-Humidity	65	6.5822	80	6.5822	96	6.5822
Subtractive-Humidity	70	3.2880	80	3.2880	95	3.2880

For each clustering technique considered, the graphs of the degree of membership of each pattern approximated to Triangular MF, Trapezoidal MF and Gaussian MF are given in Figs 4-6.

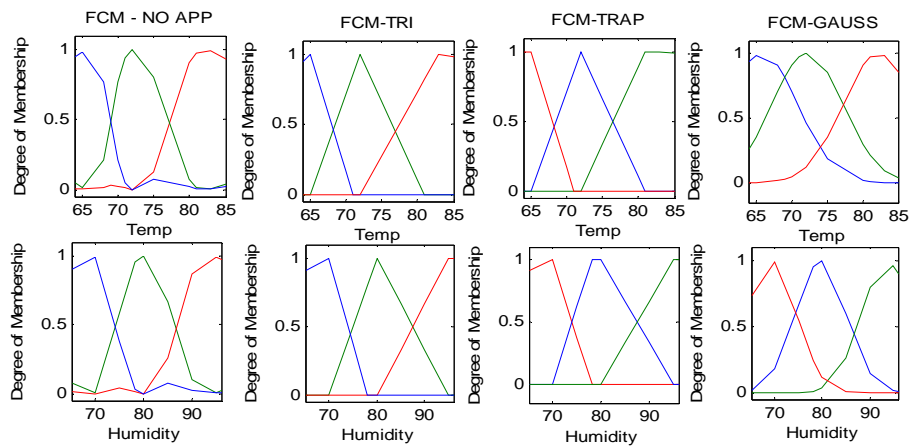


Fig. 4. FCM clustered raw data and its approximations

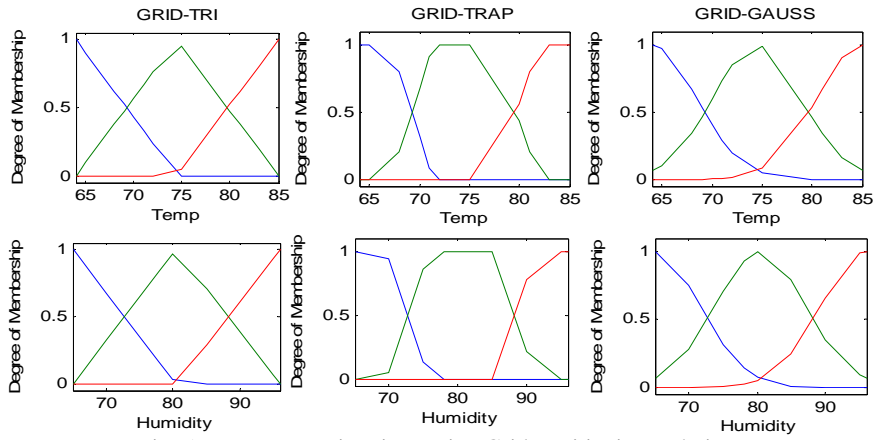


Fig. 5. Fuzzy approximations using Grid partitioning technique

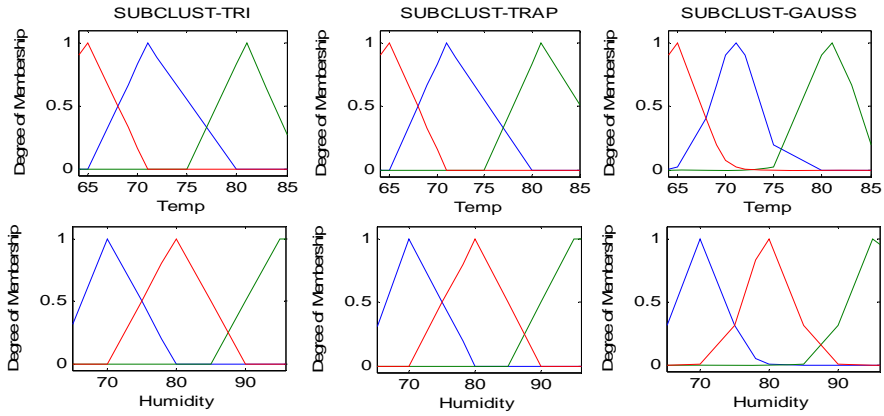


Fig. 6. Fuzzy approximations using Subtractive clustering technique

The FDT for sports dataset given in Fig. 7 is generated utilizing the values of FCM membership values directly without performing any approximation.

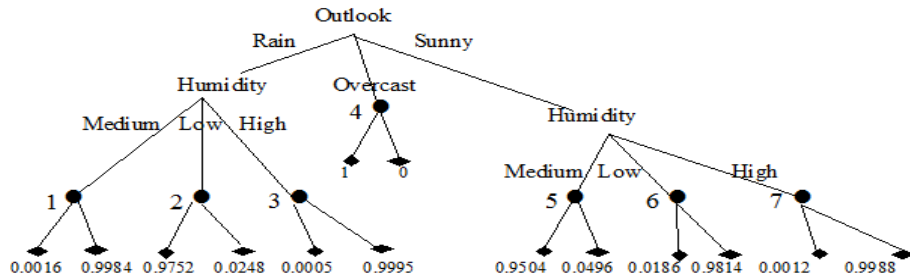


Fig. 7. Fuzzy ID3 for sports dataset

The fuzzy rules extracted from Fuzzy ID3 are:

If *Outlook* is rain and *Humidity* is medium then **Play**=0.0016 and **Don't Play**=0.9984.

If *Outlook* is rain and *Humidity* is low then **Play**=0.9752 and **Don't Play** =0.0248.

If *Outlook* is rain and *Humidity* is high then **Play**=0.0005 and **Don't Play** =0.9995.

If *Outlook* is overcast then **Play**=1 and **Don't Play** =0.

If *Outlook* is sunny and *Humidity* is medium then **Play**=0.9504 and **Don't Play** =0.0496.

If *Outlook* is sunny and *Humidity* is low then **Play**=0.0186 and **Don't Play** =0.9814.

If *Outlook* is sunny and *Humidity* is high then **Play**=0.0012 and **Don't Play** =0.9988.

The FDT with triangular approximation and trapezoidal approximation is the same as given in Fig. 7, except the certainties at the leaf nodes.

The fuzzy rules extracted from fuzzy ID3 with Triangular Approximation are:

If *Outlook* is rain and *Humidity* is medium then **Play**=1 and **Don't Play** =0.

If *Outlook* is rain and *Humidity* is low then **Play**=0 and **Don't Play** =1.

If *Outlook* is rain and *Humidity* is high then **Play**=0 and **Don't Play** =1.

If *Outlook* is overcast then **Play**=1 and **Don't Play** =0.

If *Outlook* is sunny and *Humidity* is medium then **Play**=0 and **Don't Play** =1.

If *Outlook* is sunny and *Humidity* is low then **Play**=1 and **Don't Play** =0.

If *Outlook* is sunny and *Humidity* is high then **Play**=0 and **Don't Play** =1.

The fuzzy rules extracted using fuzzy ID3 with Trapezoidal approximation are:

If *Outlook* is rain and *Humidity* is medium then **Play**=1 and **Don't Play** =0.

If *Outlook* is rain and *Humidity* is low then **Play**=0 and **Don't Play** =1.

If *Outlook* is rain and *Humidity* is high then **Play**=0 and **Don't Play** =1.

If *Outlook* is overcast then **Play**=1 and **Don't Play** =0.

If *Outlook* is sunny and *Humidity* is medium then **Play**=0 and **Don't Play** =1.

If *Outlook* is sunny and *Humidity* is low then **Play**=1 and **Don't Play** =0.

If *Outlook* is sunny and *Humidity* is high then **Play**=0 and **Don't Play** =1.

The fuzzy rules extracted using fuzzy ID3 with Gaussian approximation are:

If *Outlook* is rain and *Humidity* is low then **Play**=0.1957 and **Don't Play** =0.8043.

If *Outlook* is rain and *Humidity* is high then **Play**=0.0388 and **Don't Play** =0.9612.

If *Outlook* is rain and *Humidity* is medium then **Play**=0.9089 and **Don't Play** =0.0911.

If *Outlook* is overcast then **Play**=1 and **Don't Play** =0.

If *Outlook* is sunny and *Humidity* is low then **Play**=0.9943 and **Don't Play** =0.0057.

If *Outlook* is sunny and *Humidity* is high then **Play**=0 and **Don't Play** =1.

If *Outlook* is sunny and *Humidity* is medium and *Temp* is high then **Play**=0.0949 and **Don't Play** =0.9051.

If *Outlook* is sunny and *Humidity* is medium and *Temp* is low then **Play**=0.9318 and **Don't Play** =0.0682.

9. If *Outlook* is sunny and *Humidity* is medium and *Temp* is medium then **Play**=0.8476 and **Don't Play** =0.1524.

To study the effect of partitioning techniques and the approximation methods, several experiments were conducted. Totally 30 datasets were chosen from UCI repository [20]. For each considered dataset in Table 5, 10 fold cross validations are performed. The  $\alpha$ -cut of 0.05 is considered to reduce the fuzziness of the data. The leaf selection threshold value of 0.75 is chosen for most of the datasets. The Product-Product-Sum reasoning mechanism is used for Fuzzy decision tree inference. The percentage classification accuracy has been calculated by  $\frac{n_c}{n} \times 100\%$ ,

where  $n$  is the total number of the test patterns, and  $n_c$  is the number of test patterns classified correctly.

Table 5. Computational experimental results – classification accuracy

Dataset	FCM-Noapp	FCM-Tri	FCM-Trap	FCM-Gauss	Grid-Tri	Grid-Trap	Grid-Gauss	Sub clust-Tri	Sub clust-Trap	Sub clust-Gauss
Australian	85.74	85.74	85.74	85.74	44.12	44.12	85.59	85.74	85.74	85.74
Breast cancer	92.54	92.99	92.69	91.04	86.12	39.7	92.24	73.88	73.88	73.88
Cleveland heart disease	75.17	75.17	75.52	75.52	61.38	61.03	74.48	75.51	75.51	75.52
Pima Indian diabetes	74.74	75.13	75.13	74.87	68.42	74.08	66.97	75.00	75.92	76.05
ICU data	85.50	85.50	85.50	85.50	20.00	20.00	80.00	85.50	85.50	85.50
Iris	96.00	91.33	94.00	94.00	63.33	90.00	66.67	91.33	91.33	88.67
Omib	87.88	88.78	89.22	85.78	85.69	69.34	82.91	77.10	80.02	83.96
Planning-relax	72.22	72.22	72.22	72.22	72.22	67.78	72.22	72.22	71.66	72.22
Rice	95.00	95.00	95.00	94.00	85.00	68.00	86.00	92.00	91.00	91.00
Satellite image	73.84	72.98	75.62	74.9	75.66	70.92	70.06	73.67	73.74	72.91
Image segmentation	78.35	78.96	79.78	45.06	62.20	66.06	65.75	70.64	70.69	70.04
Thyroid disease	93.4	93.4	92.53	92.34	92.61	92.55	92.71	87.44	87.44	87.44
Thyroid Gland	87.73	86.82	87.27	72.27	82.38	87.14	80.48	70.00	70.00	70.00
Veteran Lung cancer	92.31	92.31	92.31	92.31	100	100	71.67	90.76	90.76	90.77
Wine	75.29	74.71	74.71	73.53	89.37	30.62	89.37	82.35	82.35	82.94
Glass	51.90	52.38	51.90	41.90	53.89	50.00	57.22	52.85	52.85	53.33



Table 5 (continued)

Dataset	FCM-Noapp	FCM-Tri	FCM-Trap	FCM-Gauss	Grid-Tri	Grid-Trap	Grid-Gauss	Sub clust-Tri	Sub clust-Trap	Sub clust-Gauss
Web	62.22	58.89	56.67	64.44	62.22	61.11	66.67	58.88	58.88	60.00
Breast tissue	45.00	44.00	46.00	39.00	51.25	17.5	50.00	30.00	30.00	33.00
Trans fusion	76.67	67.73	67.73	74.93	76.62	76.89	77.16	65.06	65.06	62.00
Mammo-graphic mass	82.56	82.68	82.68	79.15	79.15	79.51	79.15	79.63	79.63	79.51
Magic	78.24	80.22	80.36	74.11	74.26	77.29	74.18	77.81	77.84	77.98
Skin region segmentation	88.97	89.3	89.64	80.78	79.64	88.42	79.25	92.63	92.3	90.18
Acath	67.91	66.84	67.42	67.91	67.37	66.62	66.44	66.17	66.17	66.84
Haber-man	70.97	70.97	70.97	70.97	63.33	72.33	72.33	70.97	70.97	70.97
Liver disorder	55.71	56.86	56.57	47.14	47.35	53.52	46.47	59.14	58.86	58.85
Post operative patient	68.75	43.75	68.75	62.22	70.00	70.00	70.00	60.00	60.00	60.00
Thoracic surgery	85.1	82.97	84.89	82.76	84.89	84.89	83.82	85.31	85.1	85.68
Indian liver disorder	71.57	71.92	71.92	71.92	71.92	71.92	71.92	72.1	71.05	71.57
Stat log heart disease	77.77	71.85	77.77	71.85	48.51	48.51	77.77	68.88	68.88	68.14
Balance scale	44.83	46.5	44.83	41.29	44.83	44.83	44.83	41.12	41.12	41.12

#### 4. Inferential statistics

To compare the multiple algorithms results, the two most interesting specialized procedures for testing the significance of the differences between the multiple means are the well-known ANOVA [21] and its non-parametric counterpart, the Friedman test [22, 23]. Unfortunately, ANOVA is based on assumptions which are most probably violated when analyzing the performance of machine learning algorithms [24]. So we preferred to conduct Friedman test and its Iman Davenport extension test. But the main drawback of the Friedman and other non parametric tests like Iman Davenport test is that they can only detect significant differences over the whole multiple comparisons, being unable to locate exactly among which algorithms exists any significant difference. As an alternative, post hoc procedures, namely Nemenyi [25], Holm [26], Shaffer Statistic [27] and Bergmann-Hommel [28] tests are conducted to find among which algorithms significant differences exist. The other drawback is that the  $p$ -values given by Friedman test are not suitable for multiple comparisons. So, we prefer to choose Adjusted  $p$ -Values [29] (APVs). APVs can be compared directly with any chosen significance level. The procedure for finding the adjusted  $p$ -values is given by Joquin [30]. Considering the advantages and the conclusion given by Demsar [24] that the non-parametric tests should be preferred over the parametric ones for machine learning data, we conducted Friedman, non parametric statistical test for comparison of FDTs using KEEL software tool [31, 32] in four setups.

#### 4.1. Comparison of FDTs developed utilizing FCM values and its approximations

In this section the classification accuracies of FDT developed by directly utilizing the FCM clustered raw data without doing any approximations (FCM Noapp) and its approximations to Triangular MF (FCM Tri), Trapezoidal MF (FCM Trap) and Gaussian MF (FCM Gauss) are considered for comparison.

The  $p$ -values computed through Freidman statistics (0.00821) given in Table 6 and Iman Davenport extension ( $F_f = 4.363159$ ,  $p$ -value = 0.006535) strongly suggest the existence of significant differences among the four algorithms considered, since the  $p$ -value obtained is less than 0.05.

Table 6. Average rankings of the FDT algorithms using Fuzzy C means clustering technique

Algorithm	Ranking
FCM-Noapp	2.33
FCM-Tri	2.46
FCM-Trap	2.05
FCM-Gauss	3.15
<b>Statistic</b>	11.77
<b><math>p</math>-value</b>	0.00821

Table 7. Adjusted  $p$ -values

No	Hypothesis	Unadjusted p	pNeme	pHolm	pshaf	pBerg
1	FCM-Trap vs FCM-Gauss	0.001	0.0058	0.0058	0.0058	0.0058
2	FCM-Noapp Vs FCM- Gauss	0.0142	0.0857	0.0714	0.0428	0.0428
3	FCM-Tri vs FCM-Gauss	0.0403	0.2421	0.1614	0.1210	0.0807
4	FCM-Tri vs FCM-Trap	0.2112	1.2677	0.6338	0.6338	0.6338
5	FCM-Noapp vs FCM-Trap	0.3953	2.3719	0.7906	0.7906	0.6338
6	FCM-Noapp vs FCM- Tri	0.6891	4.1349	0.7906	0.7906	0.6891

To perform the post-hoc test, we raise the 6 hypotheses of equality among the four algorithms of FDT generated utilizing the FCM membership values and its approximations.

Table 7 lists all the hypotheses and the  $p$ -values achieved. Using a level of significance ( $\alpha$ ) of 0.10, hypotheses 1, 2 and 3 are rejected by Bergman procedure showing improvement of FCM-Trap, FCM-Noapp and FCM-Tri over FCM-Gauss. The Nemenyi, Holm and Shaffer methods reject two of the hypotheses as Bergman procedure, showing improvement of FCM-Trap and FCM-Noapp over FCM-Gauss. None of the remaining three hypotheses (4, 5 and 6) are rejected. Hence, it can be concluded that the classification accuracy of FDT developed taking FCM-Noapp membership values is equivalent to FDT developed, utilizing FCM-Trap membership values and FCM-Tri membership values, and further from the rejected hypotheses it is clear that the accuracy of FDT developed using FCM-Noapp is an improvement with respect to FDT developed utilizing FCM-Gauss membership values.

#### 4.2. Comparison of FDTs developed using grid partitioned approximations

The  $p$ -values computed through Freidman statistics (0.875173), given in Table 8 and Iman Davenport extension ( $F_f = 0.129464$ ,  $p$ -value = 0.878819) strongly suggest that there are no statistically significant differences among the FDTs generated utilizing the grid partitioned membership values.

Table 8. Average rankings of the FDT algorithms using Grid Partitioning technique

Algorithm	Ranking
Grid-Tri	2
Grid-Trap	2.066666
Grid-Gauss	1.933333
<b>Statistic</b>	0.266666
<b><math>p</math>-value</b>	0.875173

#### 4.3. Comparison of FDTs developed using Subtractive clustering and its approximations

Table 9. Average rankings of the FDT algorithms utilizing subtractive clustered data

Algorithm	Ranking
Subclust-Tri	2
Subclust-Trap	2.1
Subclust-Gauss	1.9
<b>Statistic</b>	0.6
<b><math>p</math>-value</b>	0.740818

The  $p$ -values computed with the help of Freidman statistics (0.740818), given in Table 9 and Iman Davenport extension ( $F_f = 0.292929$ ,  $p$ -value = 0.747172) strongly suggest the there are no statistically significant differences among the FDTs generated utilizing subtractive clustered membership values.

#### 4.4. Comparison of FDTs developed using different partitioning techniques

Considering FDT-Noapp as a control method from Section 4.1, 1 X N multiple comparisons are performed with all the other FDTs developed utilizing grid and subtractive approximated membership values.

Table 10. Average rankings of the FDT algorithms

Algorithm	Ranking
FCM-Noapp	2.7
Grid-Tri	4.1833
Grid-Trap	4.9
Grid-Gauss	4.15
Subclust-Tri	3.9333
Subclust-Trap	4.2333
Subclust-Gauss	3.9
<b>Statistic</b>	16.875
<b><math>p</math>-value</b>	0.009754

The  $p$ -values computed through Freidman statistics (0.009754), given in Table 10 and Iman Davenport extension ( $F_f = 3$ ,  $p$ -value: 0.008166) strongly suggest the existence of significant differences among the FDT, utilizing direct FCM membership values (FCM-Noapp) and FDT utilizing grid and subtractive clustered membership values.

Table 11. Adjusted  $p$ -values (Friedman)

Algorithm	Un Adjusted $p$	pBonf	pHolm	pHochberg	pHommel	pHolland	pRom	pFinner	pLi
Grid-Trap	0.0000	0.0004	0.0004	0.0004	0.0004	0.000	0.000	0.0004	0.000
Subclust-Trap	0.0059	0.0358	0.0298	0.0280	0.0186	0.029	0.027	0.0178	0.006
Grid-Tri	0.0078	0.0469	0.0313	0.0280	0.0234	0.030	0.027	0.0178	0.008
Grid-Gauss	0.0093	0.0560	0.0313	0.0280	0.0279	0.030	0.027	0.0178	0.009
Subclust-Tri	0.0270	0.1621	0.0540	0.0314	0.0314	0.053	0.031	0.0323	0.027
Subclust-Gauss	0.0314	0.1886	0.0540	0.0314	0.0314	0.053	0.031	0.0323	0.031

Considering the level of significance ( $\alpha$ ) being 0.10, all the six hypotheses in Table 11 are rejected by seven procedures showing that FCM-Noapp is improved in comparison with the other FDTs considered. These results support our claim that bypassing the approximation step during FDT induction is acceptable.

## 5. Conclusion

In this study we show that the FDT which utilize FCM membership values directly bypassing the approximation step, is also a good choice since it reduces the need of a domain expert to decide which parameterized function to use, it reduces the effort of choosing the appropriate cut points and other parameters for generating parameterized membership functions, it also reduces the computational complexity of the approximations to be carried out in calculating the membership values of the novel test data. The limitation of the test data that is falling within the range of the sorted training data during approximation is overcome by directly using the matrix of raw membership values. Then it is proven that the classification accuracy of FDT bypassing the approximation step is promising and acceptable.

## References

1. S a f a v i a n, S. R., D. L a n d g r e b e. A Survey of Decision Tree Classifier Methodology. – IEEE Trans SMC, Vol. **21**, 1991, No 3, pp. 660-674.
2. W e b e r, R. Fuzzy ID3: A Class of Methods for Automatic Knowledge Acquisition. – In: Proc. of International Conference on Fuzzy Logic Neural Networks, Iizuka, Japan, 1992, pp. 265-268.
3. Y u a n, Y., M. J. S h a w. Induction of Fuzzy Decision Trees. – Fuzzy Sets Syst., Vol. **69**, 1995, pp. 125-139.
4. J e n g, B., Y. M. J e n g, T. P. L i a n g. FILM: A Fuzzy Inductive Learning Method for Automated Knowledge Acquisition. – Dec. Support Syst., Vol. **21**, 1997, pp. 61-73.

5. U m a n o, M., et al. Fuzzy Decision Tree by Fuzzy ID3 Algorithm and its Application to Diagnosis Systems. – In: IEEE International Conference on Fuzzy Systems, 26-29 June 1994, pp. 2113-2118.
6. C h i a n g, I. J., J. Y. j e n H s u. Fuzzy Classification Trees for Data Analysis. – Fuzzy Sets Syst., Vol. **130**, 2002, pp. 87-99.
7. A y m e r i c h, F. X., A., J., C a b a n a s, M. E. C o m a b e l l a, M. S o b r e v i l l a, P. A. R o v i r a. Decision Tree Based Fuzzy Classifier of 1H Magnetic Resonance Spectra from Cerebrospinal Fluid Sample. – Fuzzy Sets and Systems, Vol. **170**, 2011, No 1.
8. S u g u m a r a n, N. V., K. I. R a m a c h a n d r a n. Fault Diagnosis of Roller Bearing Using Fuzzy Classifier and Histogram Features with Focus on Automatic Rule Learning. – Expert Systems with Applications, Vol. **38**, 2011, pp. 4901-4907.
9. B i s w a l, B., H. S. B e h e r a, R. B i s o i, P. K. D a s h. Classification of Power Quality Data Using Decision Tree and Chemo Tactic Differential Evolution Based Fuzzy Clustering. – Swarm and Evolutionary Computation, Vol **4**, 2012, pp. 12-24.
10. Y i n - F u, H u a n g n, C h i n g p i n g l i a n g, S i n g w u l i o u. Intron Identification Approaches Based on Weighted Features and Fuzzy Decision Trees. – Computers in Biology and Medicine, Vol. **42**, 2012, pp. 112-122.
11. N a k a n i s h i, H., I. B. T u r k s e n, M. S u g e n o. A Review and Comparison of Six Reasoning Methods. – Fuzzy Sets and Systems, Vol. **57**, 1993, pp. 257-294.
12. S u g e n o, M., T. Y a s u k a w a. A Fuzzy-Logic Based Approach to Qualitative Modelling. – IEEE Transactions on Fuzzy Systems, Vol. **1**, 1993, pp. 6-31.
13. R a j e n, B. B h a t t, S w a t h i J. N a r a y a n a n, I l a n g o P a r a m a s i v a m, M. K h a l i d. Approximating Fuzzy Membership Functions from Clustered Raw Data. – In: Proc. of IEEE International Conference on Innovations in Social and Humanitarian Engineering, 2012, pp. 487-492.
14. B e z d e k, J. C. Pattern Recognition with Fuzzy Objective Function Algorithms. – NY, Plenum Press, 1982.
15. G u i l l a u m e, S e r g e. Designing Fuzzy Inference Systems from Data: An Interpretability-Oriented Review. – IEEE Transactions on Fuzzy Systems, Vol. **9**, 2001, No 3, pp. 426-443.
16. C h i u, S. L. Fuzzy Model Identification Based on Cluster Estimation. – J. Intell. Fuzzy Systems, Vol. **2**, 1994, pp. 267-278.
17. R a j e n, B. B h a t t, M. G o p a l. FRCT: Fuzzy-Rough Classification Trees. – Pattern Anal. Applic., Vol. **11**, 2008, pp. 73-88.
18. The MathWorks, Inc. Fuzzy Logic Toolbox: User Guide (R2007b).
19. Q u i n l a n, J. R. Decision Trees and Decision Making. – IEEE Trans. SMC, Vol. **20**, 1990, No 2, pp. 339-346.
20. B l a k e, C. L., C. J. M e r z. UCI Repository of Machine Learning Databases. Irvine, CA, University of California, Department of Information and Computer Science, 1998.  
<http://www.ics.uci.edu/~mllearn/MLRepository.html>
21. F i s h e r, R. A. Statistical Methods and Scientific Inference. Second Ed. Hafner Publishing, Co., 1959, pp. 674-701.
22. F r i e d m a n, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. – Journal of the American Statistical Association, Vol. **32**, 1937, pp. 674-701.
23. F r i e d m a n, M. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. – Annals of Mathematical Statistics, Vol. **11**, 1940, pp. 86-92.
24. D e m s a r, J. Statistical Comparisons of Classifiers over Multiple Data Sets. – Journal of Machine Learning Research, Vol. **7**, 2006, pp. 1-30.
25. N e m e n y i, P. B. Distribution-Free Multiple Comparisons. Master's Thesis, Princeton University, 1963.
26. H o l m, S. A Simple Sequentially Rejective Multiple Test Procedure. – Scandinavian, Journal of Statistics, Vol. **6**, 1979, pp. 65-70.
27. S h a f f e r, J. Modified Sequentially Rejective Multiple Test Procedures. – Journal of American Statistical Association, Vol. **81**, 1986, pp. 826-831.

28. B e r g m a n n, G., G. H o m m e l. Improvements of General Multiple Test Procedures for Redundant Systems of Hypotheses. – In: P. Bauer, G. Hommel, E. Sonnemann, Eds, Multiple Hypotheses Testing, Springer, 1988, pp. 100-115.
29. W r i g h t, S. Adjusted  $p$ -Values for Simultaneous Inference. – Biometrics, Vol. **48**, 1992, pp. 1005-1013.
30. D e r r a c, J., S. G a r c i a, D. M o l i n a, F. H e r r e r a. A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. Swarm and Evolutionary Computation, 2011, pp. 3-18.
31. A l c a l a - F d e z, J., L. S a n c h e z, S. G a r c i a, M. J. d e l J e s u s, S. V e n t u r a, J. M. G a r r e l l, J. O t e r o, C. R o m e r o, J. B a c a r d i t, V. M. R i v a s, J. C. F e r n á n d e z, F. H e r r e r a. KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. – Soft Computing, Vol. **13**, 2008, No 3, pp. 307-318.
32. A l c a l a - F d e z, J., A. F e r n a n d e z, J. L u e n g o, J. D e r r a c, S. G a r c í a, L. S á n c h e z, F. H e r r e r a. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. – Journal of Multiple-Valued Logic and Soft Computing, Vol. **17**, 2011, No 2-3, pp. 255-287.