

A comparative study of three Decision Tree algorithms: ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3

Guoxiu Liang

269167

Bachelor Thesis

Informatics & Economics

Erasmus University Rotterdam

Rotterdam, the Netherlands

Augustus, 2005

Supervisors:

Dr. ir. Jan van den Berg

Dr. ir. Uzay Kaymak

Contents

Preface	4
Abstract	5
I: Introduction	6
II: Decision Tree	9
1. ID3	9
a. Entropy and Information Gain	9
b. The procedure to build the sample decision tree	14
2. Fuzzy ID3	16
a. Sample data with Fuzzy representation	16
b. Fuzzy Entropy and Information Gain	19
c. Define thresholds	21
d. The procedure to build the sample decision tree	22
e. Reasoning with Fuzzy ID3	25
3. Probabilistic Fuzzy ID3	26
a. Well-defined sample space	26
b. Statistical Fuzzy Entropy (SFE) and Information Gain	27
c. The procedure to build the sample decision tree	31

4. Comparison of ID3, FID3 and PFID3	32
a. Data representation	32
b. Termination criteria	32
c. Entropy	33
d. Reasoning	33
III: Experiment and Analysis	34
1. Purpose of the experiment	34
2. Data Set: Iris Plant Dataset	34
3. Set up	35
4. Results of PFID3 and FID3	39
5. Results of ID3	44
IV: Conclusion and Future Research	45
Bibliography	47
Appendix A	51
Dataset: <i>Iris plant dataset</i>	
Appendix B	56
Sample data and Membership functions	
Appendix C	60
Run information of WEKA	

Preface

Machine learning is an important part of my bachelor study. Decision tree (DT) is a very practical and popular approach in the machine learning domain for solving classification problems. In this thesis, a comparative study of three decision tree algorithms (ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3) is done.

I hereby extend my heartily thanks to all the teachers and friends who have provided help for this thesis:

Jan van den Berg

Uzay Kaymak

Veliana Thong

GuoXiu Liang
Rotterdam, 2006

Abstract

Decision tree learning is one of the most widely used and practical methods for inductive inference. It is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. In the past, ID3 was the most used algorithm in this area. This algorithm is introduced by Quinlan, using information theory to determine the most informative attribute. ID3 has highly unstable classifiers with respect to minor perturbation in training data. Fuzzy logic brings in an improvement of these aspects due to the elasticity of fuzzy sets formalism. Therefore, some scholars proposed Fuzzy ID3 (FID3), which combines ID3 with fuzzy mathematics theory. In 2004, another methodology Probabilistic Fuzzy ID3 (PFID3) was suggested, which is a combination of ID3 and FID3. In this thesis, a comparative study on ID3, FID3 and PFID3 is done.

Keywords

Decision tree, ID3, Fuzzy ID3, Probabilistic Fuzzy ID3, decision-making

I: Introduction

We often meet decision-making problems in our daily life or working environment. Sometimes it is very difficult for us to make good decision. In practice, we usually use our past experiences to make a decision. We can see these past experiences as a form of performing experiments to come to a correct decision. However, executing experiments costs time and money. Fortunately, the developments of computer technologies and automatic learning techniques can make this easier and more efficient. In the domain of machine learning where it always lets computers decide or come up with suggestions for the right decision, there exist many approaches of decision making techniques, such as decision trees, artificial neural networks and Bayesian learning. This thesis focuses on the decision tree approach to solve decision-making problems.

There exist many methods to do decision analysis. Each method has its own advantages and disadvantages. In machine learning, decision tree learning is one of the most popular techniques for making classifications decisions in pattern recognition.

The approach of decision tree is used in many areas because it has many advantages [17]. Compared with maximum likelihood and version spaces methods, decision tree is the quickest, especially under the condition that the concept space is large. Furthermore, it is easy to do the data preparation and to understand for non-technical people. Another advantage is that it can classify both categorical and numerical data.

The decision tree has been successfully applied to the areas of Financial Management [23] [24] [25](i.e. future exchange, stock market information, property evaluation), Business Rule Management [26](i.e. project quality analysis, product quality management, feasibility study), Banking and Insurance [27](i.e. risk forecast and evaluation), Environmental Science (i.e. environment quality appraisal, integrated resources appraisal, disaster survey) [19][21](i.e. medical decision making for making a diagnosis and selecting an appropriate treatment), and more.

A disadvantage of decision tree is its instability. Decision tree is recognized as highly unstable classifier with respect to minor perturbations in the training data [29]. The structure of the decision tree may be entirely different if some things change in the dataset. To overcome this problem, some scholars have suggested Fuzzy Decision Tree (e.g. FuzzyID3) [8][9][12][13][15] by utilizing the fuzzy set theory to describe the connected degree of attribute values, which can precisely distinguish the deference of subordinate relations between different examples and every attribute values [32].

In the beginning, Fuzzy ID3 is only an extension of the ID3 algorithm achieved by applying fuzzy sets. It generates a fuzzy decision tree using fuzzy sets defined by a user for all attributes and utilizes minimal fuzzy entropy to select expanded attributes. However, the result of this Fuzzy ID3 is poor in learning accuracy [8] [12]. To overcome this problem, two critical parameters: fuzziness control parameter θ_r and leaf decision threshold θ_n have been introduced. Besides the minimum fuzzy entropy, many different criterions have been proposed to select expanded attributes, such as the minimum classification ambiguity, the degree of the importance of attribute contribution to the classification, etc. [12]

Recently, an idea of combining fuzzy and probabilistic uncertainty has been discussed. The idea is to combine statistical entropy and fuzzy entropy into one notation termed Statistical Fuzzy Entropy (SFE) within a framework of well-defined probabilities on fuzzy events. SFE is a combination of *well-defined sample space* and fuzzy entropy. Using the notion of SFE, Probabilistic Fuzzy ID3 algorithm (PFID3) was proposed [6]. Actually, PFID3 is a special case of Fuzzy ID3. It is called PFID3 when the fuzzy partition is well defined.

The performance of the introduced approach PFID3 has never been tested before; we do not know whether its performance is better than the performance of the other two algorithms. The purpose of this thesis is to compare the performances of the algorithms ID3, FID3 and PFID3 and to verify the improvement of the proposed approach PFID3 compared with FID3.

The rest of this thesis is organized as follows: in chapter II we analyze the ID3, Fuzzy ID3 and Probabilistic Fuzzy ID3 algorithms and compare them with some simple examples. In chapter III, we set up and simulate the experiments by using Iris Plant Dataset. Finally, in the last chapter we make the conclusion after discussing and analyzing the results.

II: Decision Tree (DT)

1. ID3

Interactive Dichotomizer 3 (ID3 for short) algorithm [1] is one of the most used algorithms in machine learning and data mining due to its easiness to use and effectiveness. J. Rose Quinlan developed it in 1986 based on the Concept Learning System (CLS) algorithm. It builds a decision tree from some fixed or historic symbolic data in order to learn to classify them and predict the classification of new data. The data must have several attributes with different values. Meanwhile, this data also has to belong to diverse predefined, discrete classes (i.e. Yes/No). Decision tree chooses the attributes for decision making by using information gain (IG). [18]

a Entropy and Information Gain

ID3 [1] is the best-known algorithm for learning Decision Tree. Figure 2.1 shows a typical decision-making tree. In this example, people decide to drive the car or take the public transportation to go to work according to the weather and the traffic situation. You can find the example data in Table 2.1.

A result of the learning using ID3 tree is shown if *Traffic Jam* is *long* and *wind* is *strong*, then people will choose to take the public transportation.

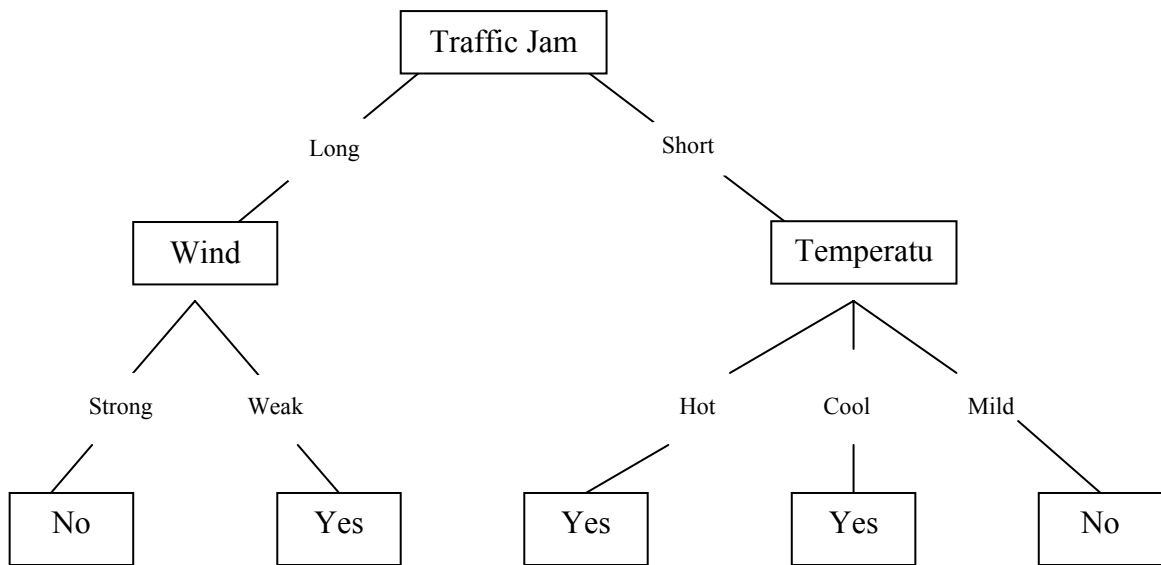


Figure 2.1 Sample Decision Tree

The basic ID3 method selects each instance attribute classification by using statistical method beginning in the top of the tree. The core question of the method ID3 is how to select the attribute of each pitch point of the tree. A statistical property called information gain is defined to measure the worth of the attribute. The statistical quantity Entropy is applied to define the information gain, to choose the best attribute from the candidate attributes. The definition of Entropy is as follows:

$$H(S) = \sum_i^N -P_i * \log_2(P_i) \quad (2.1)$$

where P_i is the ratio of class C_i in the set of examples $S = \{x_1, x_2, \dots, x_k\}$.

$$P_i = \frac{\sum_{x_k \in C_i} 1}{S} \quad (2.2)$$

For example, we assume that sample set S has 14 members altogether, including 9 positive examples and 5 negative examples. Then the Entropy of S is:

$$H(9+,5-) = -(9/14) * \log_2(9/14) - (5/14) * \log_2(5/14) = 0.940$$

Below we discuss the entropy in the special case of the Boolean classification. If all the members of set S belong to the identical kind, then the entropy is null. That means that there is no classification uncertainty.

$$H = -1 * \log_2(1) - 0 * \log_2(0) = 0.$$

If the quantity of the positive examples equals to the negative examples, then the entropy equals 1. It means maximum uncertainty.

$$H = -0.5 * \log_2(0.5) - 0.5 * \log_2(0.5) = 1$$

These results express separately that the sample set has no uncertainty (the decision is clear); or it is 100% uncertain for decision making. If the number of the positive examples is not the same as the negative examples, Entropy is situated between 0 and 1. The Figure 2.2 demonstrates the entropy relative to a Boolean classification. [7]

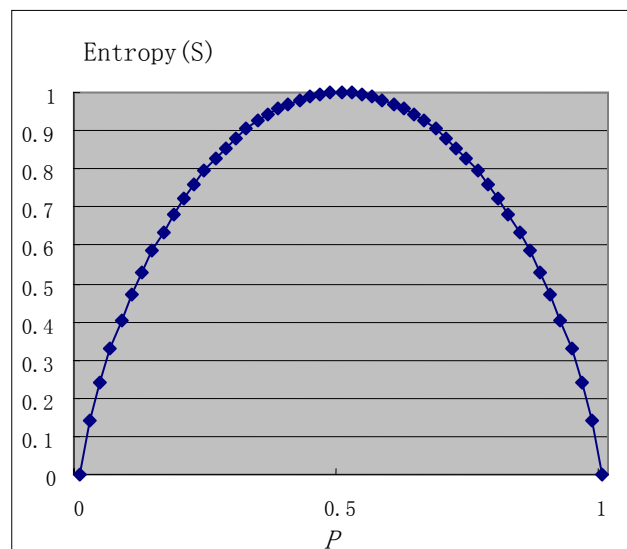


Figure 2.2: The entropy function relative to a Boolean classification, as the proportion, P , of positive examples varies between 0 and 1.

To carry on the attribute expansion, which is based on the data of this sample set, we must define a standard measure: Information Gain. An information gain of an attribute is the final information content, which is a result of the reduction of the sample set Entropy after using this attribute to divide the sample set. The definition of an information gain of an attribute A relates to the sample set S is:

$$G(S, A) = H(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} H(S_v) \quad (2.3)$$

Where: the weight $W_i = \frac{|S_v|}{|S|}$ is the ratio of the data with v attribute in the sample set.

Just like the example above, the S set [9+, 5-] contains in total 14 examples. There are 8 examples (6 positive examples and 2 negative examples) where *wind* is *weak*, and the rest with *wind* is *strong*. We can calculate the information gain of the attribute *wind* as follow:

$$S = [9+, 5-]$$

$$S(\text{weak}) = [6+, 2-]$$

$$S(\text{strong}) = [3+, 3-]$$

$$\begin{aligned} G(S, \text{wind}) &= H(S) - (8/14)H(\text{Weak}) - (6/14)H(\text{Strong}) \\ &= 0.940 - (8/14)*0.811 - (6/14)*1.0 \\ &= 0.048 \end{aligned}$$

Using the same principle, we may calculate the information gains of attributes.

Temperature and Traffic-Jam:

$$\begin{aligned} G(S, Tem) &= H(S) - (4/14)H(hot) - (6/14)H(mild) - (4/14) * H(cool) \\ &= 0.940 - (4/14) * 1.0 - (6/14) * 0.918 - (4/14) * 0.811 \\ &= 0.029 \end{aligned}$$

$$\begin{aligned} G(S, Traffic) &= H(S) - (7/14)H(Long) - (7/14)H(Short) \\ &= 0.940 - (7/14)*0.985 - (7/14)*0.592 \\ &= 0.151 \end{aligned}$$

As a result, we may choose the highest information content, which belongs to the variable *Traffic-Jam*, to carry on the first expansion of the classification.

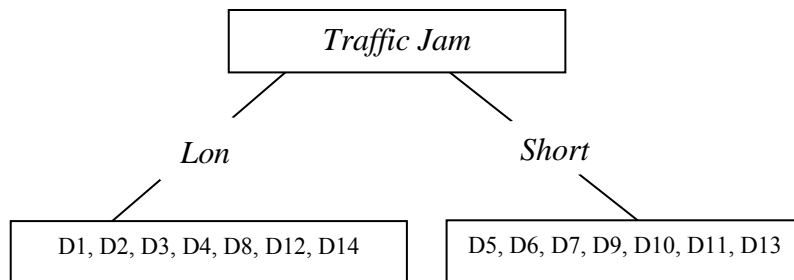


Figure 2.3: the first classification according to the highest Gain *Traffic-Jam*

b The procedure to build the decision tree

We take the original samples as the root of the decision tree. As the result of the calculation above, the attribute *Traffic Jam* is used to expand the tree. Two sub-nodes are generated. The left and the right sub-node of the root separately contain the samples with the attribute value *Long* and *Short*. Left sub-node = [D1, D2, D3, D4, D8, D12, D14], right sub-node = [D5, D6, D7, D9, D10, D11, D13].

We then delete the attribute *Traffic-Jam* of the samples in these sub-nodes and compute the Entropy and the Information Gain to expand the tree using the attribute with highest gain value. We repeat this process until the Entropy of the node equals null. At that moment, the node cannot be expanded anymore because the samples in this node belong to the same class. In the end we get the decision tree as in Figure 2.1

Following is the ID3 algorithm [7]:

ID3 (*Examples, Target-attribute, Attributes*)

Examples are the training examples. Target-attribute is the attribute whose value is to be predicted by the tree. Attributes are a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given examples.

- Create a *Root* node for the tree
- If all *Examples* are positive, return the single-node tree *Root*, with label = +
- If all *Examples* are negative, return the single-node tree *Root*, with label = -
- If *Attributes* is empty, return the single-node tree *Root*, with label = most common value of *Target-attribute* in *Examples*
- Otherwise Begin
 - $A \leftarrow$ the attribute from *Attributes* that best* classifies *Examples*
 - The decision attribute for *Root* $\leftarrow A$
 - For each possible value, v_i , of A ,
 - Add a new tree branch below *Root*, corresponding to the test $A = v_i$
 - Let $Examples_{v_i}$ be the subset of examples that have value v_i for A
 - If $Examples_{v_i}$ is empty
 - Then below this new branch add a leaf node with label = most common value on value of *Target-attribute* in *Examples*
 - Else below this new branch add the sub-tree

ID3 (*Examples, Target-attribute, Attributes*- $\{A\}$)
- End
- Return *Root*

*The best attribute is the one with highest information gain

2. Fuzzy ID3

Fuzzy decision tree is an extension of classical decision tree and an effective method to extract knowledge in uncertain classification problems. It applies the fuzzy set theory to represent the data set and combines tree growing and pruning to determine the structure of the tree.

a Sample data with Fuzzy representation

In general, there exist two different kinds of attributes: discrete and continuous. Many algorithms require data with discrete value. It is not easy to replace a continuous domain with a discrete one. This requires some partition and clustering. It is also very difficult to define the boundary of the continuous attributes. For example, how do we define whether the traffic-jam is long or short? Can we say that the traffic-jam of 3 km is long, and 2.9 km is short? Can we say it is cool when the temperature is 9, and it is mild for 10? Therefore, some scholars quote the fuzzy concept in the method ID3, substitute the sample data with the fuzzy expression and form the fuzzy ID3 method. Below is the example of the fuzzy representation for the sample data.

We know that in most areas, the space of the temperature factor x is approximately between $[-50, 50]$. Then the membership functions of the fuzzy set *hot*, *mild* and *cool* μ_x may be defined separately as:

Attribute *Temperature*:

$$\mu_c(x) = \begin{cases} 1 & x < 0 \\ 1 - x/15 & 0 \leq x \leq 15 \\ 0 & x > 15 \end{cases} \quad (2.4)$$

$$\mu_m(x) = \begin{cases} 0 & x < 5 \\ x/15 - 1/3 & 5 \leq x < 20 \\ 1 & 20 \leq x < 30 \\ -x/5 + 7 & 30 \leq x < 35 \\ 0 & x > 35 \end{cases} \quad (2.5)$$

$$\mu_h(x) = \begin{cases} 0 & x < 25 \\ x/10 - 2.5 & 25 \leq x \leq 35 \\ 1 & x > 35 \end{cases} \quad (2.6)$$

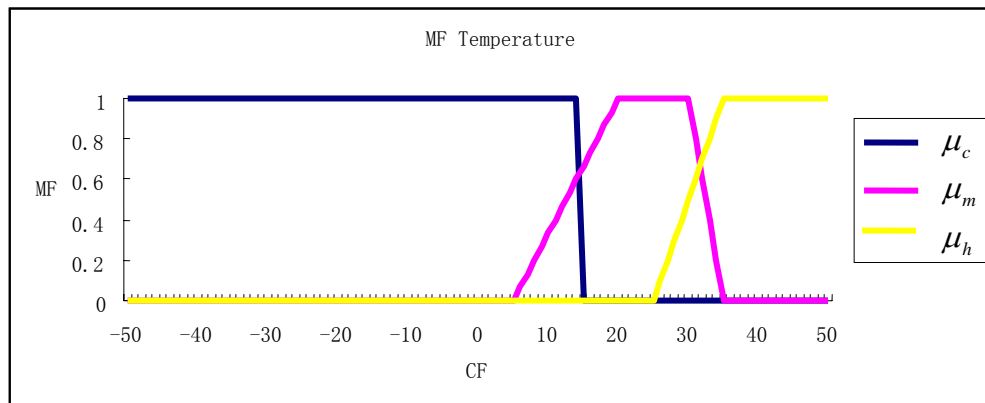


Figure 2.4: graphic representations of the membership function of *Temperature*.

Also we can define the membership functions of *Wind* and *Traffic-Jam*.

Attribute *Wind*:

$$\mu_w(x) = \begin{cases} 1 & x < 3 \\ 2.5 - x/2 & 3 \leq x \leq 5 \\ 0 & x > 5 \end{cases} \quad (2.7)$$

$$\mu_{st}(x) = \begin{cases} 0 & x < 3 \\ x/5 - 0.6 & 3 \leq x \leq 8 \\ 1 & x > 8 \end{cases} \quad (2.8)$$

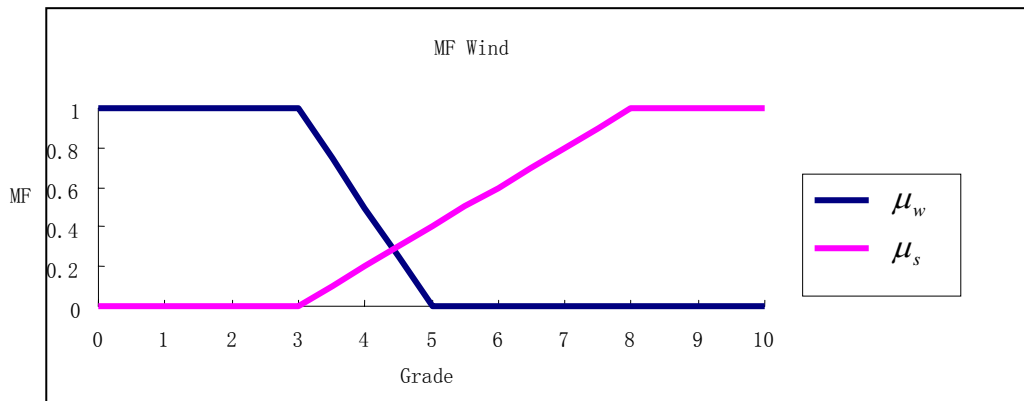


Figure 2.5: graphic representations of the membership function of *Wind*.

Attribute *Traffic-Jam*:

$$\mu_{sh}(x) = \begin{cases} 1 & x < 3 \\ 1.5 - x/6 & 3 \leq x \leq 9 \\ 0 & x > 9 \end{cases} \quad (2.9)$$

$$\mu_l(x) = \begin{cases} 0 & x < 5 \\ x/10 - 0.5 & 5 \leq x \leq 15 \\ 1 & x > 15 \end{cases} \quad (2.10)$$

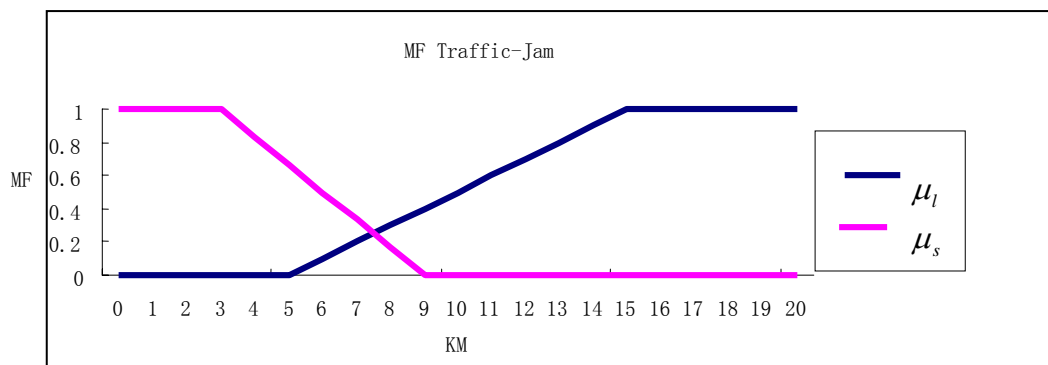


Figure 2.6: graphic representations of the membership function of *Traffic-Jam*.

As example above, we have partitioned the sample set into different intervals. The partition is complete (each domain value is belong to at lease one subset) and inconsistent (a domain value can be found in more than one subset).

Example *Traffic-Jam*:

If *traffic-jam* is 3 km, the value of the MF *Long* is null.

If *traffic-jam* is 3 km, the value of the MF *Short* is one.

b Fuzzy Entropy and Information Gain

Next, we have to calculate the fuzzy Entropy and Information Gain of the fuzzy data set to expand the tree.

In this case, we get the same result of the entropy of the as ID3

$$H_f(S) = H_s(S) = 0.940$$

The formulas of the entropy for the attributes and the Information Gain are a little bit different because of the data fuzzy expression. Their definitions are defined as follow respectively with the assumption dataset $S = \{x_1, x_2, \dots, x_j\}$:

$$H_f(S, A) = -\sum_{i=1}^C \frac{\sum_j \mu_{ij}}{S} \log_2 \frac{\sum_j \mu_{ij}}{S} \quad (2.11)$$

$$G_f(S, A) = H_f(S) - \sum_{v \in A} \frac{|S_v|}{|S|} * H_{fs}(S_v, A) \quad (2.12)$$

where: μ_{ij} is the membership value of the j^{th} pattern to the i^{th} class.

$H_f(S)$ presents the entropy of the set S of training examples in the node.

$|S_v|$ is the size of the subset $S_v \subseteq S$ of training examples x_j with v attribute.

$|S|$ presents the size of set S .

Now we do the calculations on the example set:

- Information Gain of attribute *Temperature*

$$H_f(Tem, c) = -4/6 * \log_2 4/6 - 2/6 * \log_2 2/6 = 0.918$$

$$H_f(Tem, m) = -4.94/7.41 * \log_2 4.94/7.41 - 2.47/7.41 * \log_2 2.47/7.41 = 0.918$$

$$H_f(Tem, c) = -1.2/2.7 * \log_2 1.2/2.7 - 1.5/2.7 * \log_2 1.5/2.7 = 0.991$$

$$\begin{aligned} G_f(S, Tem) &= 0.940 - 6/16.11 * 0.918 - 7.41/16.11 * 0.918 - 2.7/16.11 * 0.991 \\ &= 0.0098 \end{aligned}$$

- Information Gain of attribute *Wind*

$$H_f(W, w) = -6.5/9 * \log_2 6.5/9 - 2.5/9 * \log_2 2.5/9 = 0.852$$

$$H_f(W, s) = -1.1/2.1 * \log_2 1.1/2.1 - 1/2.1 * \log_2 1/2.1 = 0.998$$

$$G_f(S, W) = 0.940 - 9/11.1 * 0.852 - 2.1/11.1 * 0.998 = 0.06$$

- Information Gain of attribute *Traffic-Jam*

$$H_f(T, l) = -0.96/2.09 * \log_2 0.96/2.09 - 1.13/2.09 * \log_2 1.13/2.09 = 0.995$$

$$H_f(T, s) = -6.01/7.81 * \log_2 6.01/7.81 - 1.8/7.81 * \log_2 1.8/7.81 = 0.779$$

$$G_f(S, T) = 0.940 - 2.09/9.9 * 0.995 - 7.81/9.9 * 0.779 = 0.1154$$

The same as the result of ID3, the information gain of the attribute *Traffic Jam* has the highest value. We use it to expand the tree.

c Define thresholds

If the learning of FDT stops until all the sample data in each leaf node belongs to one class, it is poor in accuracy. In order to improve the accuracy, the learning must be stopped early or termed pruning in general. As a result, two thresholds are defined [8].

- Fuzziness control threshold θ_r

If the proportion of a data set of a class C_k is greater than or equal to a threshold θ_r , stop expanding the tree.

For example: if in sub-dataset the ratio of class 1 is 90%, class 2 is 10% and θ_r is 85%, then stop expanding.

- Leaf decision threshold θ_n

If the number of a data set is less than a threshold θ_n , stop expanding.

For example, a data set has 600 examples where θ_n is 2%. If the number of samples in a node is less than 12 (2% of 600), then stop expanding.

The level of these thresholds has great influences on the result of the tree. We define them in different levels in our experiment to find optimal values.

Moreover, if there are no more attributes for classification, the algorithm does not create a new node.

- d The procedure to build the fuzzy decision tree: [8]

Create a *Root* node that has a fuzzy set of all data with membership value 1. With the result of the calculation above, we use the attribute *Traffic-Jam* to expand the tree. Generate two sub-nodes with the examples, where the membership values at these sub-nodes are the product of the original membership values at *Root* and the membership values of the attribute *Traffic-Jam*. The example is omitted if its membership value is null.

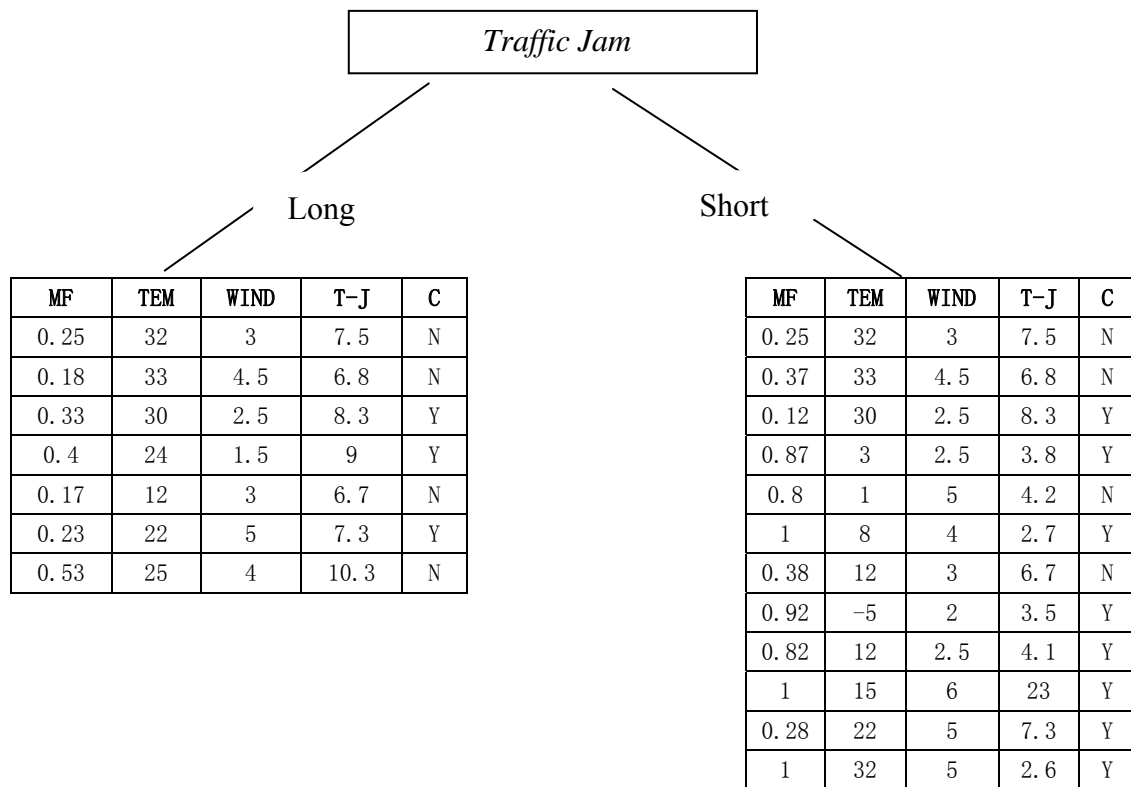
For example, for the left sub-node with attribute value *Long*, the membership value of data 1(D1) μ_l equals to 0.25. The new membership value of D1 in this node is 0.25. Below is the calculation:

$$\mu_{new} = \mu_l * \mu_{old} = 0.25 * 1 = 0.25$$

See the rest result in figure 2.7.

Next we have to calculate the proportion of the class C_k . It is the quotient of the sum of membership values of class C_k to the sum of all the membership values. For example, in the left sub-node, the proportion of class N is $1.13/2.09=54\%$. The number of the dataset is 7. After that we compare the proportion and the number of dataset with θ_r and θ_n . If they are smaller than θ_r and θ_n and if there are also attributes for classification, then we go further to create a new node. Repeat these processes until the stop conditions defined in b) are satisfied.

For example, the proportion of the class Y in the right sub-node is 77%. If the user-defined fuzzy control parameter is 70%, we stop expanding this node. In this case, it means that if *traffic Jam* is *short*, the probabilities of Not-driving and Well-driving are 23% and 77% respectively.



Class membership N: $1.13/2.09=54\%$

Class membership Y: $0.96/2.09=46\%$

N: $1.8/7.81=23\%$

Y: $6.01/7.81=77\%$

Figure 2.7 generated sub-tree of FID3

Calculation of the class membership:

- Left-node:

$$C_N = 0.25 + 0.18 + 0.17 + 0.53 = 1.13$$

$$C_Y = 0.33 + 0.4 + 0.23 = 0.96$$

Total membership value: $1.13 + 0.96 = 2.09$

Proportion of the class N: 54%. Proportion of the class Y: 46%

- Right-node:

$$C_n = 0.25 + 0.37 + 0.8 + 0.38 = 1.8$$

$$C_Y = 0.12 + 0.87 + 1.0 + 0.92 + 0.82 + 1.0 + 0.28 + 1.0 = 6.01$$

Total membership value: $1.8 + 6.01 = 7.81$

Proportion of the class N: 23%. Proportion of the class Y: 77%.

In this case, if θ_r is 95%, then both of the nodes expand; if θ_r is 75%, the left-node expands and the right-node stops expanding.

Fuzzy ID3 Algorithm [8]

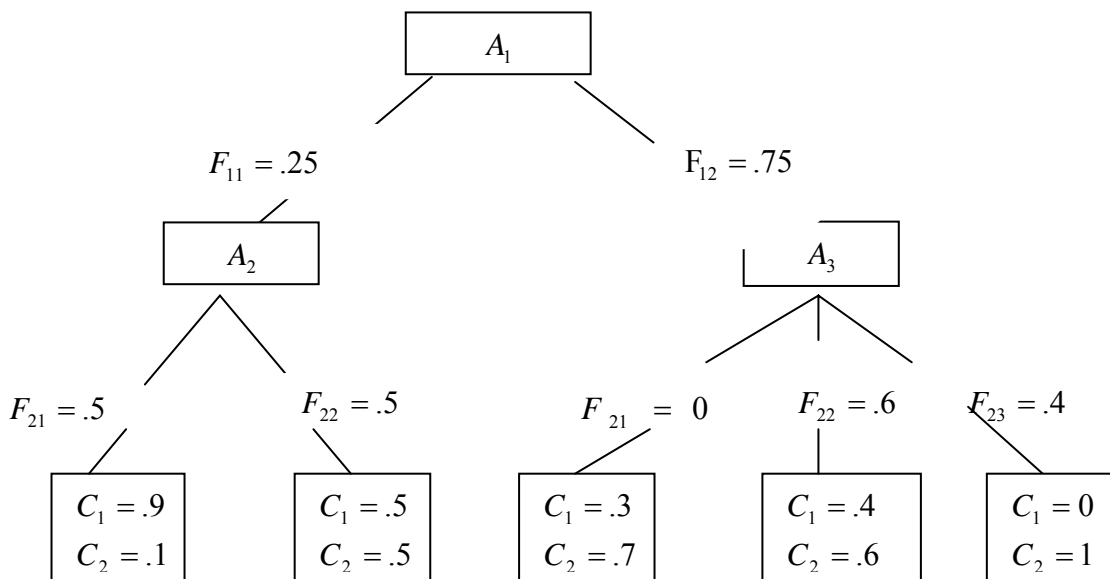
- 1 Create a *Root* node that has a set of fuzzy data with membership value 1
- 2 If a node t with a fuzzy set of data D satisfies the following conditions, then it is a leaf node and assigned by the class name
 - The proportion of a class C_k is greater than or equal to θ_r ,

$$\frac{|D^{C_k}|}{|D|} \geq \theta_r$$
 - the number of a data set is less than θ_n
 - there are no attributes for more classifications
- 3 If a node D does not satisfy the above conditions, then it is not a leaf-node. And a new sub-node is generated as follows:
 - For A_i 's ($i=1, \dots, L$) calculate the information gain $G(2.8)$, and select the test attribute A_{\max} that maximizes them.
 - Divide D into fuzzy subsets D_1, \dots, D_m according to A_{\max} , where the membership value of the data in D_j is the product of the membership value in D and the value of $F_{\max,j}$ of the value of A_{\max} in D .
 - Generate new nodes t_1, \dots, t_m for fuzzy subsets D_1, \dots, D_m and label the fuzzy sets $F_{\max,j}$ to edges that connect between the nodes t_j and t
 - Replace D by D_j ($j=1, 2, \dots, m$) and repeat from 2 recursively.

e Reasoning with Fuzzy ID3

We must start reasoning from the top node (*Root*) of the fuzzy decision tree. Repeat testing the attribute at the node, branching an edge by its value of the membership function (μ) and multiplying these values until the leaf node is reached. After that we multiply the result with the proportions of the classes in the leaf node and get the certainties of the classes at this leaf node. Repeat this action until all the leaf nodes are reached and all the certainties are calculated. Sum up the certainties of the each class respectively and choose the class with highest certainty [8].

We present the sample calculation by means of Figure 2.8. Each of tree node and leaf node represent the value of the MF of the attribute at the node and the proportion of each class at the node respectively. By using method called $X-X+$ [8], we have the result that the sample belongs to C_1 and C_2 with probabilities 0.355 and 0.645 respectively. These probabilities are complement.



Figures 2.8 Representing of fuzzy reasoning

$$C_1 = .25 * .5 * .9 + .25 * .5 * .5 + .75 * 0 * .3 + .75 * .6 * .4 + .75 * .4 * 0 = .355$$

$$C_2 = .25 * .5 * .1 + .25 * .5 * .5 + .75 * 0 * .7 + .75 * .6 * .6 + .75 * .4 * 1 = .645$$

$$C_1 + C_2 = 1$$

3. Probabilistic Fuzzy ID3

a Well-defined sample space

Zadeh defines the probability of a fuzzy event as: the probability of a fuzzy event A , given probability density function $f(x)$, is found by taking the mathematical expectation of the membership function. [6]

$$P_r(A) = \int_{-\infty}^{\infty} \mu_A(x) f(x) dx = E(\mu_A(x)) \quad (2.13)$$

For a discrete sample with outcomes x_1, x_2, \dots , this formula reduces to

$$P_r(A) = \sum_i \mu_A(x_i) f(x_i) = E(\mu_A(x)) \quad (2.14)$$

As we know, the sum of the probability of events in different situations, which occurs at the same time, is 1. Consider throwing a dice as an example of discrete events. The possible outcomes are 1, 2, 3, 4, 5 and 6. The probability of each situation is equal to 1/6, and the sum is 1. In the other words, regardless which point we may obtain, it is only possible to obtain one result a time. It is absolutely impossible to obtain two or more results simultaneously. If fuzzy events A_1, A_2, \dots, A_n form a proper fuzzy partition in sample space X , it implies that

$$\forall x : \sum_{i=1}^N \mu_{A_i}(x) = 1 \quad (2.15)$$

Then the sum of the probabilities of the fuzzy events equals to one. All data points have equal weight.

It is the same as the non-fuzzy events we mentioned above.

$$\sum_{i=1}^N P_r(A_i) = 1 \quad (2.16)$$

The fuzzily partitioned sample having this property is termed *well-defined sample space* [6].

b Statistical Fuzzy Entropy (SFE)

In the previous chapter we have already introduced the fuzzy events with *well-defined sample space* and the Fuzzy Entropy. We combine the *well-defined sample space* and fuzzy entropy into statistical fuzzy entropy within a framework of well-defined probabilities on fuzzy events. We may get the formula of the Statistical Fuzzy Entropy (SFE).

$$H_{sf} = -\sum_{c=1}^C E(\mu_{A_c}(x) \log_2(E(\mu_{A_c}(x)))) \quad (2.17)$$

Now we apply the SFE into the Statistical Fuzzy decision trees.

We generalize the Statistical Information Gain to the Statistical Fuzzy Information Gain by replacing the Entropy with SFE in formula (2.3).

$$G(S, A) = H_{sf}(S) - \sum_i \frac{|S_i|}{|S|} H_{sf}(S_i) \quad (2.18)$$

To satisfy the condition of *well-defined sample space*, the sum of the membership function must be equal to 1. We fix the membership functions of the Fuzzy D3. We see that the membership grade of each event equals to 1 in figure 2.9-2.11.

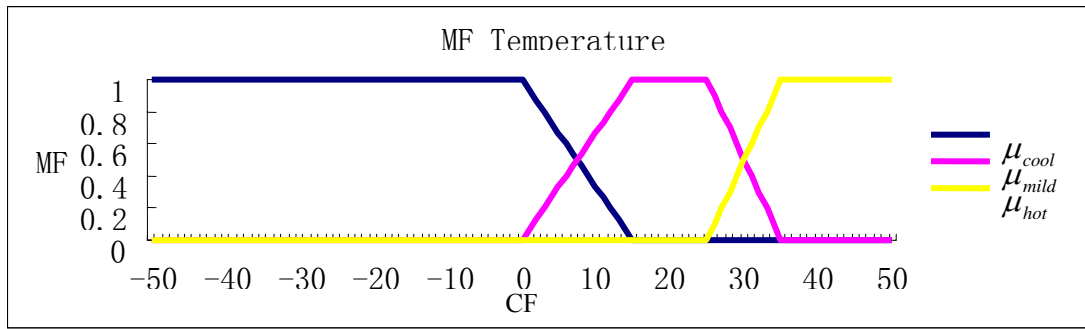


Figure 2.9: membership function of *Temperature*.*

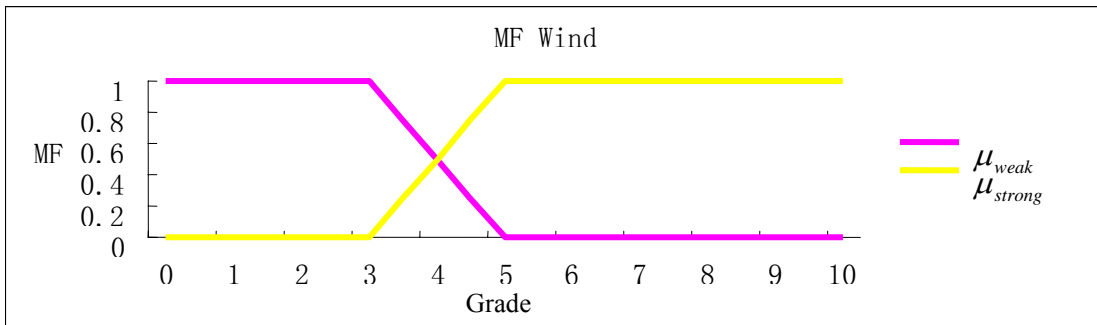


Figure 2.10: membership function of *Wind*.*

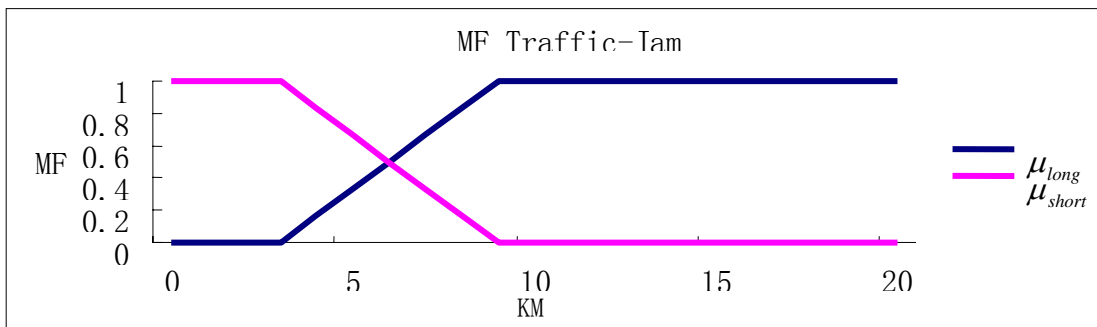


Figure 2.11: membership function of *Traffic-Jam*.*

* See the formulas of the membership function in the appendix

In order to calculate the Entropies of *hot*, *mild* and *cool*, we firstly calculate

$$\begin{aligned} Pr(h, Y) &= Pr(\text{Temperature} = \text{hot and Car driving} = \text{Yes}) \\ &= \frac{\sum_{\text{yes}} \mu_{\text{hot}}}{\sum \mu_{\text{hot}}} = \frac{1.2}{1.2 + 1.5} = 0.44 \end{aligned}$$

Similarly, we get $Pr(h, N) = 1.5/2.7 = 0.56$,

$$Pr(m, Y) = 5.35/7.7 = 0.69, Pr(m, N) = 2.37/7.7 = 0.31,$$

$$Pr(c, Y) = 2.47/3.6 = 0.69, Pr(c, N) = 1.13/3.6 = 0.31$$

By using (2.5), we can calculate the entropy.

$$H_{sf}(S_{\text{hot}}, \text{Tem}) = -0.44 * \log_2 0.44 - 0.56 * \log_2 0.56 = 0.99$$

$$H_{sf}(S_{\text{mild}}, \text{Tem}) = -0.69 * \log_2 0.69 - 0.31 * \log_2 0.31 = 0.89$$

$$H_{sf}(S_{\text{cool}}, \text{Tem}) = -0.31 * \log_2 0.31 - 0.69 * \log_2 0.69 = 0.89$$

Finally, by using (2.7), we find the Information Gain of the fuzzy variable *Temperature*:

$$G_{sf}(S, \text{Tem}) = 0.940 - \frac{2.7}{14} * 0.99 - \frac{7.7}{14} * 0.89 - \frac{3.6}{14} * 0.89 = 0.03$$

Similarly, we also calculate the Information Gain of the fuzzy variables *Wind* and *Traffic*.

$$G_{sf}(S, \text{Wind}) = 0.940 - \frac{9.25}{14} * 0.879 - \frac{4.75}{14} * 0.998 = 0.02$$

$$G_{sf}(S, \text{Traffic}) = 0.940 - \frac{6.2}{14} * 0.999 - \frac{7.8}{14} * 0.779 = 0.06$$

If we go further with the research, we find that the sub-tree of PFID3 is similar to FID3 after the first expansion. See figure 2.12 below:

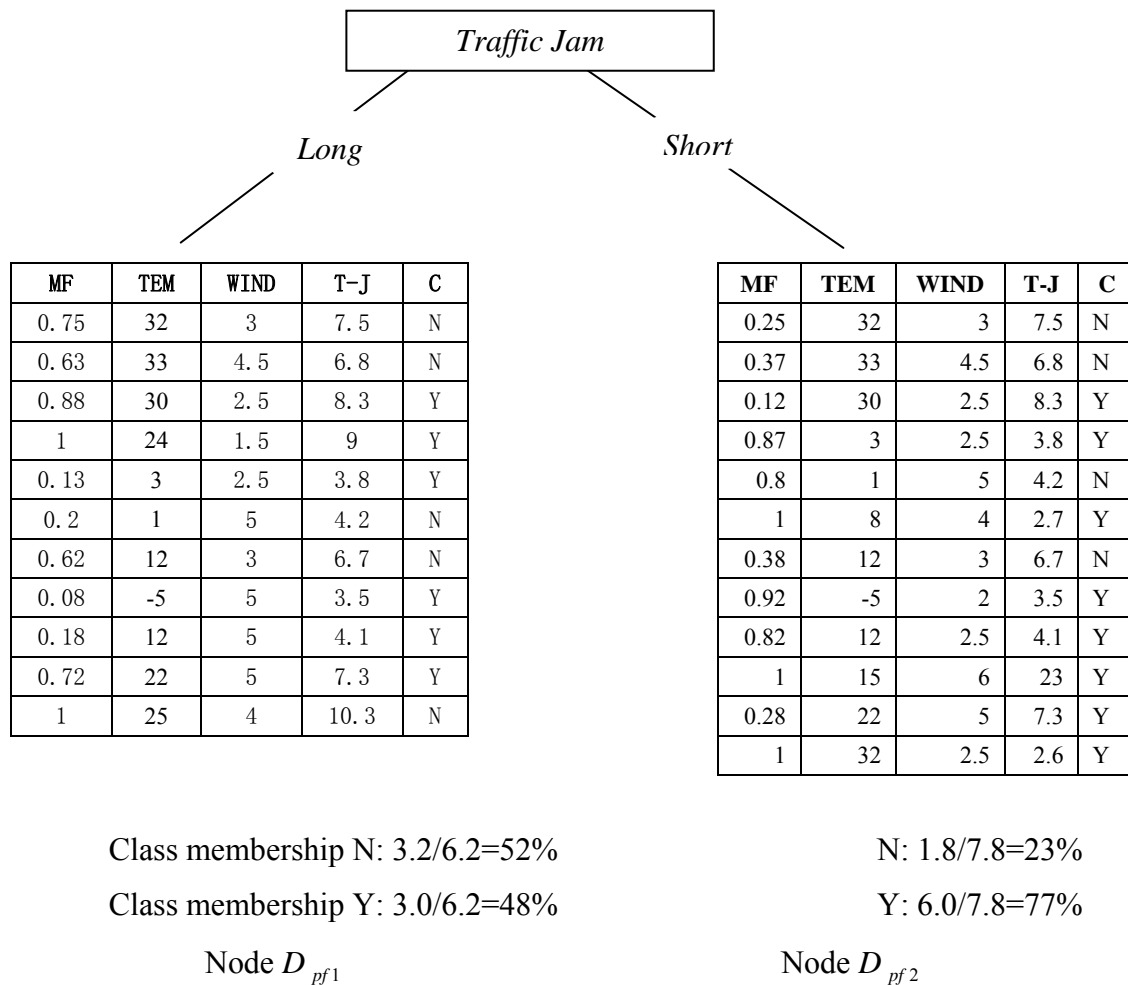


Figure 2.12 generated sub-tree of PFID3

The calculations of the class membership values in left-node are as below:

$$\begin{aligned} \text{Total-MF} &= 0.75+0.63+\dots+1.0 = 6.2 \\ \text{Total-MF(N)} &= 0.75+0.63+0.2+0.62+1.0=3.2 \\ \text{Total-MF(Y)} &= 0.88+1.0+0.13+0.08+0.18+0.72=3.0 \\ \text{MF}(C_N) &= 52\% \\ \text{MF}(C_Y) &= 48\% \end{aligned}$$

c The procedure to build the sample decision tree

Except for the fuzzy representations of the input sample, the rest of the processes of PFID3 are the same as FID3.

We can see that both of the right sub-nodes are the same because of the identical membership functions of the value *Short*. Let us analyze the left sub-nodes. The node of FID3 is called D_{f1} and that of PFID3 is called D_{pf1} . There are in total 11 samples in D_{pf1} and 7 in D_{f1} . Because there are 4 membership values in D_{f1} which is equal to null, FID3 has passed over them. In this case, the probabilities of these 4 samples are less than one. Also, it is possible that the probabilities of some examples are greater than one. It means that there is less than one situation for event *A* to happen in the same time. Just like the example ‘dice’ which we have explained above, it is not possible to have a possibility less than one. By applying *well-defined sample spaces*, this kind of problem does not occur.

Probabilistic Fuzzy ID3 Algorithm

- | |
|---|
| <ol style="list-style-type: none">1 Create a <i>Root</i> node that has a set of fuzzy data with membership value 1 that fits the condition of <i>well-defined sample space</i>.2 Execute the Fuzzy ID3 algorithm from step 2 to end. |
|---|

4. Comparing the algorithms among ID3, FID3 and PFID3

Because FID3 and PFID3 are based on ID3, these three methodologies have similar algorithms. However, there also exist some differences.

a Data representation:

The data representation of ID3 is crisp while for FID3 and PFID3, they are fuzzy, with continuous attributes. Moreover, the membership functions of PFID3 must satisfy the condition of *well-defined sample space*. The sum of all the membership values for all data value x_i must be equal to 1.

b Termination criteria:

ID3: if all the samples in a node belong to one class or in other words, if the entropy equals to null, the tree is terminated. Sometimes, people stop learning when the proportion of a class at the node is greater than or equal to a predefined threshold. This is called pruning. The pruned ID3 tree stops early because the redundant branches have been pruned.

FID3 & PID3: there are three criteria's.

- 1) If the proportion of the dataset of a class is greater than or equal to a threshold θ_r ,
- 2) If the number of a data set is less than another threshold θ_n ,
- 3) If there are no more attributes at the node to be classified

If one of these three criteria's is fulfilled, the learning is terminated.

c Entropy

- ID3:

$$H(S) = \sum_i^N -P_i * \log_2(P_i)$$

- FID3 & PFID3:

$$H_{sf} = -\sum_{c=1}^C E(\mu_{A_c}(x)) \log_2(E(\mu_{A_c}(x)))$$

d Reasoning

The reasoning of the classical decision tree begins from the root node of the tree, and then branch one edge to test the attribute of the sub node. Repeat the testing until the leaf node is reached. The result of ID3 is the class attached to the leaf node.

The reasoning of the fuzzy decision trees is different. It does not branch one edge, but all the edges of the tree. It begins from the root node through the branches to the leaf nodes until all the leaf nodes have been tested. Each leaf node has various proportions of all the classes. In other words, each leaf node has own certainties of the classes. The result is the aggregation of the certainties at all the leaf nodes.

Let us see what the difference is among the results of the experiment in the next chapter.

III: Experiment and Analysis

1. Purpose of the experiment

In order to compare the performance of the three algorithms, we build the FID3 and PFID3 tree models using Matlab and the ID3 model using WEKA. All the decision trees are pruned in order to get the accurate comparing results.

2. Data set: Iris Plant Database

The Iris plant dataset is applied to the experiment. The dataset is created by R.A. Fisher and perhaps it is the best-known database found in the pattern recognition literature. The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. There are in total 4 numerical attributes and no missing value in the dataset [11].

Attribute	Description
A1	Sepal length in cm
A2	Sepal width in cm
A3	Petal length in cm
A4	Petal width in cm

Table 3.1: Descriptions of the attributes

Class	Description
C1	Iris Setosa
C2	Iris Versicolour
C3	Iris Virginica

Table 3.2: Descriptions of the classes

3. Setup

There is an internal classifier of ID3 in WEKA. We can do the experiment easily without having to program the ID3 by ourselves again. Therefore, we only focus on the setup of PFID3 and FID3.

We program the FID3 and PID3 algorithms by using Matlab 7.1. Matlab is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation [16]. Matlab also has a special toolbox to handle the fuzzy programming: Fuzzy Logic Toolbox. That is why we choose it.

First we have to normalize the dataset and then find the cluster centers of each class by using Matlab internal clustering function. The number of the clusters has a great effect on the result of the experiment. How to choose the correct number is not discussed here, and will be done in the future research. In this case, we cluster the dataset into 3 crowds because the dataset contains 3 classes. After the clustering, we get the 3 sets of cluster centers. Based on these cluster centers, we partition the dataset into 3 fuzzy classes (Low, Mid and High) by using the membership functions as below.

Membership functions for method FID3:

- Low: using Z-shaped curve membership function [14].

This spline-based function of x is so named because of its Z-shape. The parameters a and b locate the extremes of the sloped portion of the curve as given by:

$$MF = \begin{cases} 1 & x \leq a \\ 1 - 2\left(\frac{x-a}{b-a}\right)^2 & a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{b-x}{b-a}\right)^2 & \frac{a+b}{2} \leq x \leq b \\ 0 & x \geq b \end{cases} \quad (3.1)$$

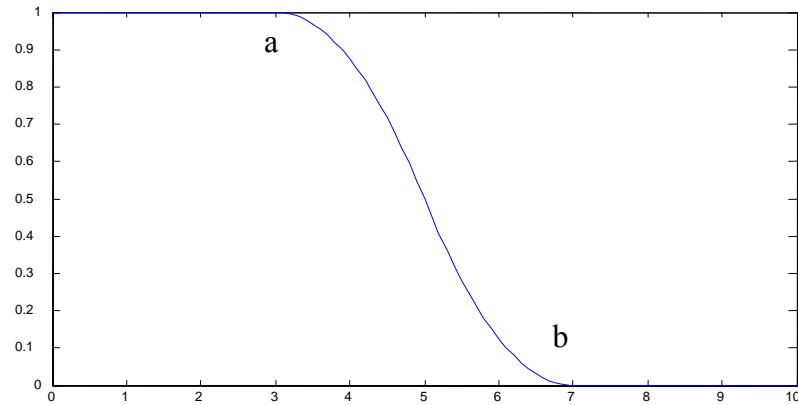


Figure 3.1: Z-shape curve

- Mid: using Gaussian curve membership function [21].

The symmetric Gaussian function depends on two parameters σ and c as given by:

$$MF = e^{\frac{-(x-c)^2}{2\sigma^2}} \quad (3.2)$$

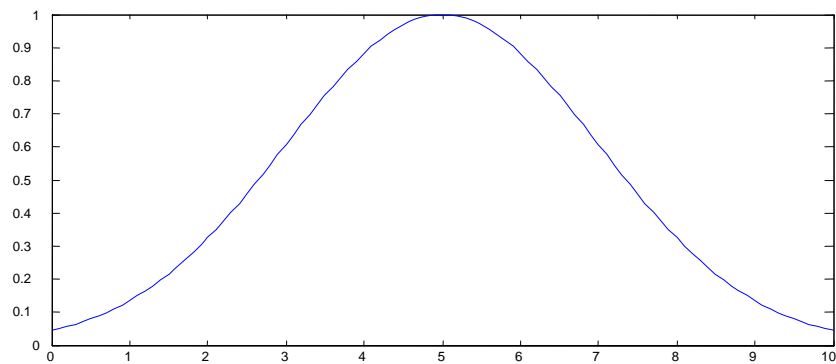


Figure 3.2: Gaussian curve

- High: using S-shaped curve membership function [20].

This spline-based curve is a mapping on the vector x , and is so named because of its S-shape. The parameters b and c locate the extremes of the sloped portion of the curve.

$$MF = \begin{cases} 0 & x \leq b \\ 2\left(\frac{x-b}{c-b}\right)^2 & b \leq x \leq \frac{b+c}{2} \\ 1 - 2\left(\frac{c-x}{c-b}\right)^2 & \frac{b+c}{2} \leq x \leq c \\ 1 & x \geq c \end{cases} \quad (3.3)$$

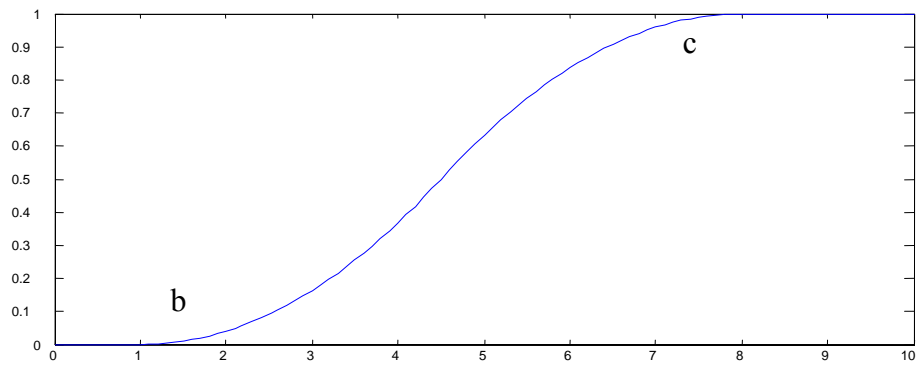


Figure 3.3: S-shape curve

Membership functions under condition *well-defined sample space* for method PFID3:

- Low: using Z-shape membership function [14].
The membership function is identical to equation (3.1).
- Mid: using Π -shaped curve membership function [28].
This spline-based curve is so named because of its Π shape. This membership function is evaluated at the points determined by the vector x . The parameters a and c are located at the both feet of the curve, while b is located at its top. See Figure 3.4.

$$MF = \begin{cases} 0 & x \leq a \\ 2\left(\frac{x-a}{b-c}\right)^2 & a \leq x \leq \frac{a+b}{2} \\ 1-2\left(\frac{b-x}{b-a}\right)^2 & \frac{a+b}{2} \leq x \leq b \\ 1-2\left(\frac{x-b}{c-b}\right)^2 & b \leq x < \frac{b+c}{2} \\ 2\left(\frac{c-x}{c-b}\right)^2 & \frac{b+c}{2} \leq x \leq c \\ 0 & x \geq c \end{cases} \quad (3.4)$$

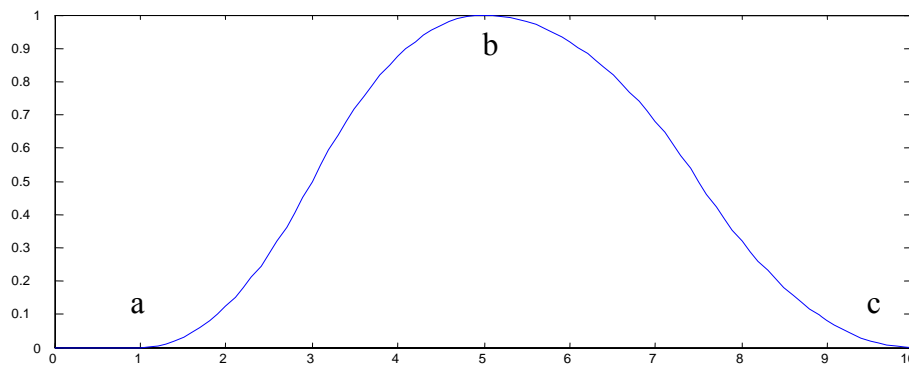


Figure 3.4: Π -shape curve

- High: using S-shape membership function [20].
The membership function is identical to equation (3.3).

4. Results of FID3 and PFID3:

- Result of clustering

Below is the table of the cluster centers.

	A1	A2	A3	A4
a	5.0036	3.4030	1.4850	0.25154
b	5.8890	2.7612	4.3640	1.39730
c	6.7749	3.0524	5.6467	2.05350

Table 3.3: cluster centers

We see that the order of center points of A1, A3 and A4 is ascend, whereas A2 is not monotonic. In this case, we have to adjust the order of the cluster centers (a, b and c) when we apply the membership function to attribute A2. We get Table 3.4.

	A1	A2	A3	A4
a	5.0036	2.7612	1.4850	0.25154
b	5.8890	3.0524	4.3640	1.39730
c	6.7749	3.4030	5.6467	2.05350

Table 3.4: adjusted cluster centers which applied in MF functions

Table 3.5 presents the cluster centers and the standard deviation (SD) of the second cluster, which are applied to equation 3.2. In this case, it is not possible to calculate the fuzzy standard deviation (FSD) because the membership value of each data point needed for calculating the FSD, is not defined yet. Therefore, we partition the dataset average into 3 parts and calculate the SD of the second parts.

	A1	A2	A3	A4
c	5.88900	3.0524	4.36400	1.39730
σ	0,51617	0.3138	0.46991	0.19775

Table 3.5: adjusted cluster centers and standard deviation of the second cluster

- Result of Fuzzification

By filling the cluster centers (a, b, c) of Table 3.4 in the predefined membership functions (equations 3.1, 3.3, 3.4) and applying the centers(c) and standard deviation (σ) of Table 3.5 to the equation 3.2, we get the graphics of the membership functions.

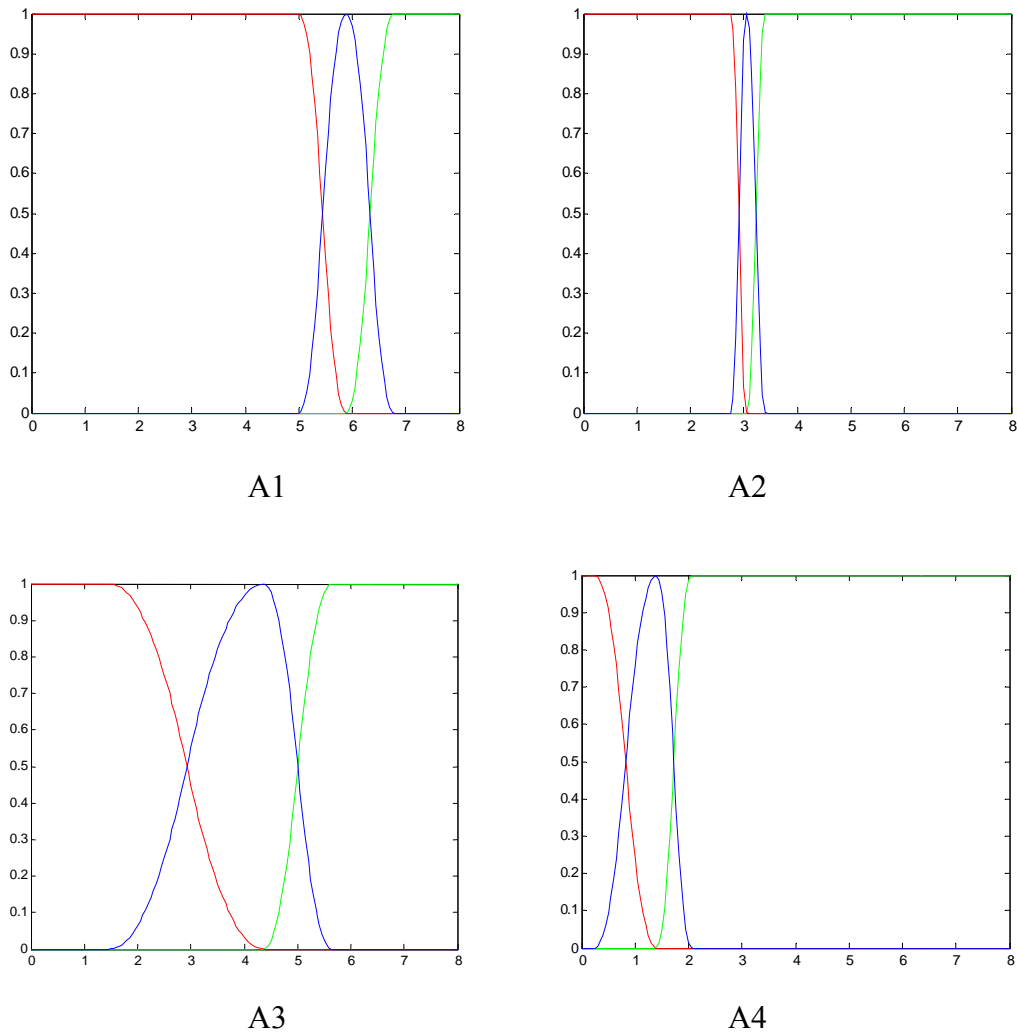


Figure 3.5 Graphic of MF **with** *well-defined sample space* by using cluster centers

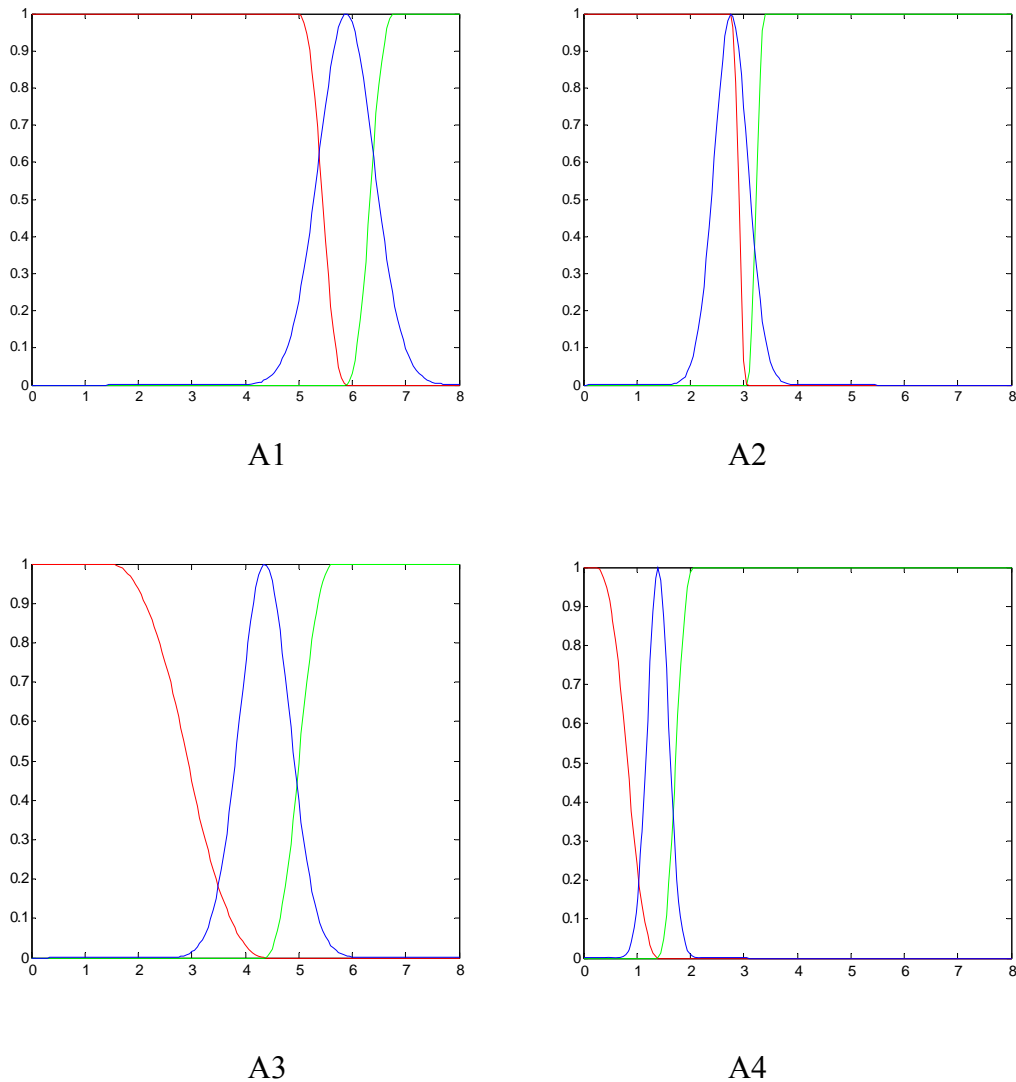


Figure 3.6 Graphic of MF **without** *well-defined sample space* by using cluster centers

Figure 3.5 and 3.6 are the graphic fuzzy representations of the attributes A1, A2, A3 and A4 under condition with and without *well-defined sample space* respectively. Y-axis is the value of the membership function, X-axis is the value of data x_i . We see that the red and green lines (MF of High and Low) of Figure 3.5 and 3.6 are the same, only the blue lines (MF of Mid) are different. It is correct according to the predefined MF. Figure 3.5 shows that under the condition *well-defined sample space*, the sum of all membership values ($\sum \mu_i$) equals to 1 for each point.

- Results of the experiment

We define the splitting size as $2/3$. The two thirds of the dataset are used as training dataset to train the algorithm. The rest of one third is the validation set in order to test the performance of the methodology. The experiments of FID3 and PFID3 are repeated 9 times by using 9 pairs of thresholds. In order to get the best performance and the global view of the average performance, we build 100 decision trees in each experiment with different random sample sets and tests them with the test set.

The results of the experiments are given in Table 3.6 according to different threshold parameters θ_r and θ_n . The program is run 100 times. *Mean* is the average hit rate performance of the experiments, in which each experiment uses different random selected training set and test set. *SD* is the standard deviation. *Min* and *Max* are respectively the minimum and maximum performance value of the experiments.

The result shows that PFID3 performs better than FID3 in classification for all situations. The best performance of PFID3 is 100% under all kinds of conditions. In general, the method PFID3 performs well, smoothly and stably. The performances of PFID3 increase when the fuzzy control parameter decreases parameter increases. If we use the same θ_n and decrease θ_r , the performance increases. The best performance of PFID3 is 95% with $\theta_r=0.80$, $\theta_n=0.1$.

The best performance of FID3 is 91.4% with $\theta_r=0.80$, $\theta_n=0.1$. We can see that the performances are monotonic. Therefore, it is difficult to find a rule between the thresholds and the performances. In general, FID3 performs better when $\theta_r=0.80$.

$\theta_r=0.95$								
θ_n	PFID3				FID3			
	Mean	SD	Min	Max	Mean	SD	Min	Max
0.4	<u>0.934</u>	0.029	0.860	1.000	0.777	0.074	0.540	0.900
0.2	0.932	0.042	0.780	1.000	<u>0.779</u>	0.075	0.540	0.940
0.1	0.930	0.043	0.700	1.000	0.767	0.080	0.540	0.920
$\theta_r=0.90$								
	PFID3				FID3			
	Mean	SD	Min	Max	Mean	SD	Min	Max
0.4	<u>0.945</u>	0.034	0.860	1.000	0.771	0.086	0.580	0.940
0.2	0.940	0.031	0.840	1.000	0.760	0.084	0.460	0.940
0.1	0.942	0.026	0.880	1.000	0.769	0.076	0.580	0.940
$\theta_r=0.80$								
	PFID3				FID3			
	Mean	SD	Min	Max	Mean	SD	Min	Max
0.4	0.947	0.024	0.880	1.000	0.910	0.053	0.560	0.980
0.2	0.949	0.026	0.880	1.000	0.898	0.071	0.560	0.980
0.1	<u>0.950</u>	0.028	0.860	1.000	0.914	0.044	0.620	0.980

Table 3.6 Performances of PFID3 and FID3

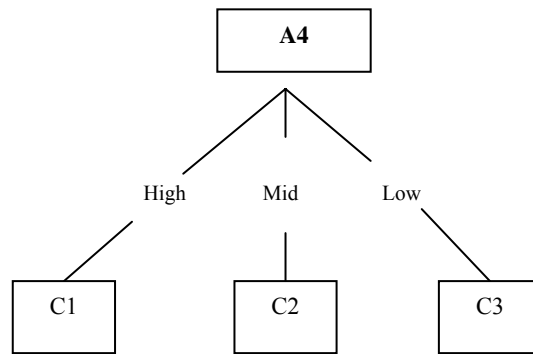


Figure 3.7: Sample structure of FID3 and PFID3 with $\theta_r=0.80$, $\theta_n=0.1$

5. Results of ID3

The dataset is also split into 2 sub dataset, with $2/3$ as the training set and $1/3$ as the test set. The termination criterion of the pruned ID3 tree is the minimum number of the data at the left-node. In order to compare the results with FID3 and PFID3, we use the same level as that of the leaf decision parameter θ_n .

Below is the table summary with the best results:

Min No of instances	The same as θ_n	Hit rate
5	0.1	0.922
10	0.2	0.922
20	0.4	0.941

Tale 3.7: Summary table of ID3

Below is the tree structure of ID3 under all the conditions mentioned above.

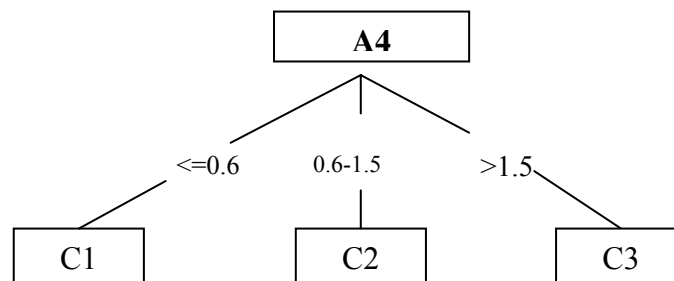


Figure 3.8: Tree structure of ID3

IV: Conclusion and Future Research

Table 4.1 summarizes the average performances of FID3 and PFID3 and the best one from ID3. In general, PFID3 performs the best, follows by ID3 and finally FID3. The best hit-rates are 94.7%, 94.9% and 95% under condition that $\theta_n=0.1$, $\theta_n=0.2$ and $\theta_n=0.4$ respectively.

θ_n	PFID3	FID3	ID3
0.1	0.947	0.910	0.922
0.2	0.949	0.898	0.922
0.4	0.950	0.914	0.941

Table 4.1: Summary table of performances

First, we compare the results of PFID3 and ID3, the performance of PFID3 is always better. For $\theta_n=0.1$, ID3 is 0.025 worse than PFID3, while for $\theta_n=0.2$, it is 0.027 and for $\theta_n=0.4$, it is 0.009. Table 4.2 shows the percentage of how much ID3 is worse than PFID3. We conclude that applying the *well-defined sample space* to the fuzzy partition have a positive effect on the performance.

θ_n	PFID3	ID3	%
0.1	-	-0.025	2.64%
0.2	-	-0.027	2.85%
0.4	-	-0.009	0.95%

Table 4.2: Summary table of difference between PFID3 and ID3

Now, we only compare PFID3 and FID3. PFID3 performs much better than FID3 under all conditions. The main difference between the learning PFID3 and FID3 is the *well-defined sample space*. The weight of the each data point of PFID3 is equal to one. Therefore, the data reacts on the learning with the same weight; each data has the same contribution to reasoning. On the contrary, the data point of FID3 can be overweight or underweight. Thus, the learning is inaccurate due to the imbalanced weight of the data. In other words, the origin of the better accuracy of PFID3 is the weight consistency of the data. We consider this phenomenon as the *effect of well-defined sample space*. But we need more evidences to support this viewpoint. In the further research, more experiments will be executed and evaluated.

The leaf decision threshold is very important to the performance of the learning. In general, the performance increases along with the increasing of leaf decision threshold. This happens because when the leaf decision threshold increases, the learning has pruned the redundant branches. However, if the threshold increases too much, it causes *underfitting*. Finding the best leaf decision threshold will be done in the future research.

The partitions of the fuzzy data do significantly have effect on the performance of the learning. In this thesis, we use the cluster centers to do the fuzzy partition. We find that the number of the clusters determines the number of the membership functions, which then affects the fuzzy partition. How to define the number of the clusters is not discussed in this thesis, it is left for the future research.

The definitions of the membership functions also have a great effect on the performance of the learning. In this case, we just randomly choose the internal membership function of Matlab as the membership function of the dataset. How to find the best parameters and define the best membership function could further be researched in the future.

Bibliography

- [1] Quinlan, J. R. *Induction of Decision Trees*. Machine Learning, vol. 1, pp. 81-106, 1986.
- [2] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, P. J. *Classification and regression trees*. Belmont, CA: Wadsworth International Group, 1984.
- [3] Jang, J.-S.R., Sun, C.-T., Mizutani, E. *Neuro-Fuzzy and Soft Computing – A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1997.
- [4] Brodley, C. E., & Utgoff, P.E. *Multivariate decision trees*. Machine Learning, vol. 19, pp. 45-77, 1995
- [5] Quinlan, J. R. *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.
- [6] vd Berg, J. and Uzay Kaymak. *On the Notion of Statistical Fuzzy Entropy*. Soft Methodology and Random Information Systems, Advances in Soft Computing, pp. 535–542. Physica Verlag, Heidelberg, 2004.
- [7] Mitchell, T.M. *Machine Learning*, McGraw-Hill, 1997.
- [8] M. Umamo, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, J. Kinoshita. *Fuzzy Decision Trees by Fuzzy ID3 algorithm and Its Application to Diagnosis Systems*. In Proceedings of the third IEEE Conference on Fuzzy Systems, vol. 3, pp. 2113-2118, Orlando, June, 1994.

- [9] Chang, R.L.P., Pavlidis, T. *Fuzzy decision tree algorithms*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 7 (1), pp. 28-35, 1977.
- [10] Han, J., Kamber, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
- [11] Fisher, R.A. *Iris Plant Dataset*. UCI Machine Learning Repository.
www.ics.uci.edu/~mllearn/MLRepository.html (on 15 Nov. 2005)
- [12] Wang, X.Z., Yeung, D.S., Tsang, E.C.C. *A Comparative Study on Heuristic Algorithms for Generating Fuzzy Decision Trees*. IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 31(2), pp. 215-226, 2001.
- [13] Janikow, C.Z. *Fuzzy decision trees: issues and methods*. IEEE Transactions on Systems, Man, and Cybernetics, Part B, vol. 28(1), pp. 1-14, 1998.
- [14] *Fuzzy Logical Toolbox: zmf*. Matlab Help. The MathWorks, Inc. 1984-2004.
- [15] Peng, Y., Flach, P.A. *Soft Discretization to Enhance the Continuous Decision Tree Induction*. Integrating Aspects of Data Mining, Decision Support and Meta-learning, pp. 109-118, 2001.
- [16] *What is Matlab?* Matlab Help. The Math Works, Inc. 1984-2004.
- [17] *Decision Tree*. Wikipedia, the free encyclopedia.
http://en.wikipedia.org/wiki/Decision_tree (on 15 Nov. 2005)
- [18] *The ID3 Algorithm*. Department of Computer and Information Science and Engineering of University of Florida.
<http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm> on 15. Nov. 2005.

- [19] Le Loet, X., Berthelot, J.M. Cantagre, A, Combe, B., de Bandt, M., Fautrel, B., Filpo, R.M., Liote, F., Maillefert, J.F., Meyer, O., Saraux, A., Wendling, D. and Guillemin, F. *Clinical practice decision tree for the choice of the first disease modifying antirheumatic drug for every early rheumatoid arthritis: a 2004 proposal of the French Society of Rheumatology.* June 2005.
- [20] *Fuzzy Logical Toolbox: smf.* Matlab Help. The Math Works, Inc. 1984-2004.
- [21] *Fuzzy Logical Toolbox: gaussmf.* Matlab Help. The Math Works, Inc. 1984-2004.
- [22] *Stakeholder pensions and decision trees.* Financial Services Authority (FSA) fact sheets. April 2005.
- [23] *The Federal Budget Execution Process Decision Tree.* Know Net.
<http://www.knownet.hhs.gov> (on 15 Nov. 2005)
- [24] *Data Mining for Profit.* Rosella Data mining & Database Analytics
<http://www.roselladb.com/> (on 15 Nov. 2005)
- [25] *Neusoft Financial Industry Solution.* Neusoft Ltd., Com 2003.
<http://www.neusoft.com/framework/browse/F119F39EE1@0000/?folderId=2142>
(on 15 Nov. 2005)
- [26] *Business rule management.* Flagleader Ltd., Com 2004.
<http://www.flagleader.com> (on 15 Nov. 2005)
- [27] Sahinoglu, M. *Security Meter: A Practical Decision-Tree Model to Quantify Risk.* IEEE Security and Privacy, vol. 03(3), pp. 18-24, May/June, 2005.
- [28] *Fuzzy Logical Toolbox: pimf.* Matlab Help. The Math Works, Inc. 1984-2004.
- [29] Olaru C., Wehenkel L. *A complete fuzzy decision tree technique.* Fuzzy set and systems, pp. 221-254, 2003.

- [29] Boyan, X. and Wehenkel, L. *Automatic induction of fuzzy decision trees and its application to power system security assessment*. Fuzzy set and Systems, vol 102(1), pp. 3-19, 1999.
- [30] Marsala, C. and Bouchon-Meunier¹, B. *From Fuzzy Questionnaires to Fuzzy Decision Trees 30 years of Research in Fuzzy Learning*.
- [31] Marsala, C. *Application of Fuzzy Rule Induction to Data Mining*. In Proc. of the 3rd Int. Conf. FQAS'98, Roskilde (Denmark), LNAI nr. 1495, pp. 260-271, 1998.
- [32] *Soft Decision Tree*.

Appendix A

Dataset: *Iris Plants Database*

1. Title: Iris Plants Database
 2. Sources:
 - (a) Creator: R.A. Fisher
 - (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
 - (c) Date: July, 1988
 3. Number of Instances: 150 (50 in each of three classes)
 4. Number of Attributes: 4 numeric, predictive attributes and the class
 5. Attribute Information:
 1. sepal length in cm
 2. sepal width in cm
 3. petal length in cm
 4. petal width in cm
 5. class:
 - Iris Setosa
 - Iris Versicolour
 - Iris Virginica
 6. Missing Attribute Values: None
- Summary Statistics:
- | | Min | Max | Mean | SD | Class Correlation |
|---------------|-----|-----|------|------|-------------------|
| Sepal length: | 4.3 | 7.9 | 5.84 | 0.83 | 0.7826 |
| Sepal width: | 2.0 | 4.4 | 3.05 | 0.43 | -0.4194 |
| Petal length: | 1.0 | 6.9 | 3.76 | 1.76 | 0.9490 (high!) |
| Petal width: | 0.1 | 2.5 | 1.20 | 0.76 | 0.9565 (high!) |
7. Class Distribution: 33.3% for each of 3 classes.

5,1	3,5	1,4	0,2	1
4,9	3	1,4	0,2	1
4,7	3,2	1,3	0,2	1
4,6	3,1	1,5	0,2	1
5	3,6	1,4	0,2	1
5,4	3,9	1,7	0,4	1
4,6	3,4	1,4	0,3	1
5	3,4	1,5	0,2	1
4,4	2,9	1,4	0,2	1
4,9	3,1	1,5	0,1	1
5,4	3,7	1,5	0,2	1
4,8	3,4	1,6	0,2	1
4,8	3	1,4	0,1	1
4,3	3	1,1	0,1	1
5,8	4	1,2	0,2	1
5,7	4,4	1,5	0,4	1
5,4	3,9	1,3	0,4	1
5,1	3,5	1,4	0,3	1
5,7	3,8	1,7	0,3	1
5,1	3,8	1,5	0,3	1
5,4	3,4	1,7	0,2	1
5,1	3,7	1,5	0,4	1
4,6	3,6	1	0,2	1
5,1	3,3	1,7	0,5	1
4,8	3,4	1,9	0,2	1
5	3	1,6	0,2	1
5	3,4	1,6	0,4	1
5,2	3,5	1,5	0,2	1
5,2	3,4	1,4	0,2	1
4,7	3,2	1,6	0,2	1
4,8	3,1	1,6	0,2	1
5,4	3,4	1,5	0,4	1
5,2	4,1	1,5	0,1	1
5,5	4,2	1,4	0,2	1
4,9	3,1	1,5	0,1	1
5	3,2	1,2	0,2	1
5,5	3,5	1,3	0,2	1
4,9	3,1	1,5	0,1	1
4,4	3	1,3	0,2	1
5,1	3,4	1,5	0,2	1
5	3,5	1,3	0,3	1
4,5	2,3	1,3	0,3	1
4,4	3,2	1,3	0,2	1
5	3,5	1,6	0,6	1
5,1	3,8	1,9	0,4	1

4,8	3	1,4	0,3	1
5,1	3,8	1,6	0,2	1
4,6	3,2	1,4	0,2	1
5,3	3,7	1,5	0,2	1
5	3,3	1,4	0,2	1
7	3,2	4,7	1,4	2
6,4	3,2	4,5	1,5	2
6,9	3,1	4,9	1,5	2
5,5	2,3	4	1,3	2
6,5	2,8	4,6	1,5	2
5,7	2,8	4,5	1,3	2
6,3	3,3	4,7	1,6	2
4,9	2,4	3,3	1	2
6,6	2,9	4,6	1,3	2
5,2	2,7	3,9	1,4	2
5	2	3,5	1	2
5,9	3	4,2	1,5	2
6	2,2	4	1	2
6,1	2,9	4,7	1,4	2
5,6	2,9	3,6	1,3	2
6,7	3,1	4,4	1,4	2
5,6	3	4,5	1,5	2
5,8	2,7	4,1	1	2
6,2	2,2	4,5	1,5	2
5,6	2,5	3,9	1,1	2
5,9	3,2	4,8	1,8	2
6,1	2,8	4	1,3	2
6,3	2,5	4,9	1,5	2
6,1	2,8	4,7	1,2	2
6,4	2,9	4,3	1,3	2
6,6	3	4,4	1,4	2
6,8	2,8	4,8	1,4	2
6,7	3	5	1,7	2
6	2,9	4,5	1,5	2
5,7	2,6	3,5	1	2
5,5	2,4	3,8	1,1	2
5,5	2,4	3,7	1	2
5,8	2,7	3,9	1,2	2
6	2,7	5,1	1,6	2
5,4	3	4,5	1,5	2
6	3,4	4,5	1,6	2
6,7	3,1	4,7	1,5	2
6,3	2,3	4,4	1,3	2
5,6	3	4,1	1,3	2
5,5	2,5	4	1,3	2
5,5	2,6	4,4	1,2	2

6,1	3	4,6	1,4	2
5,8	2,6	4	1,2	2
5	2,3	3,3	1	2
5,6	2,7	4,2	1,3	2
5,7	3	4,2	1,2	2
5,7	2,9	4,2	1,3	2
6,2	2,9	4,3	1,3	2
5,1	2,5	3	1,1	2
5,7	2,8	4,1	1,3	2
6,3	3,3	6	2,5	3
5,8	2,7	5,1	1,9	3
7,1	3	5,9	2,1	3
6,3	2,9	5,6	1,8	3
6,5	3	5,8	2,2	3
7,6	3	6,6	2,1	3
4,9	2,5	4,5	1,7	3
7,3	2,9	6,3	1,8	3
6,7	2,5	5,8	1,8	3
7,2	3,6	6,1	2,5	3
6,5	3,2	5,1	2	3
6,4	2,7	5,3	1,9	3
6,8	3	5,5	2,1	3
5,7	2,5	5	2	3
5,8	2,8	5,1	2,4	3
6,4	3,2	5,3	2,3	3
6,5	3	5,5	1,8	3
7,7	3,8	6,7	2,2	3
7,7	2,6	6,9	2,3	3
6	2,2	5	1,5	3
6,9	3,2	5,7	2,3	3
5,6	2,8	4,9	2	3
7,7	2,8	6,7	2	3
6,3	2,7	4,9	1,8	3
6,7	3,3	5,7	2,1	3
7,2	3,2	6	1,8	3
6,2	2,8	4,8	1,8	3
6,1	3	4,9	1,8	3
6,4	2,8	5,6	2,1	3
7,2	3	5,8	1,6	3
7,4	2,8	6,1	1,9	3
7,9	3,8	6,4	2	3
6,4	2,8	5,6	2,2	3
6,3	2,8	5,1	1,5	3
6,1	2,6	5,6	1,4	3
7,7	3	6,1	2,3	3
6,3	3,4	5,6	2,4	3

6,4	3,1	5,5	1,8	3
6	3	4,8	1,8	3
6,9	3,1	5,4	2,1	3
6,7	3,1	5,6	2,4	3
6,9	3,1	5,1	2,3	3
5,8	2,7	5,1	1,9	3
6,8	3,2	5,9	2,3	3
6,7	3,3	5,7	2,5	3
6,7	3	5,2	2,3	3
6,3	2,5	5	1,9	3
6,5	3	5,2	2	3
6,2	3,4	5,4	2,3	3
5,9	3	5,1	1,8	3

Appendix B

Sample data and Membership functions

Day	Temperature	Wind	Traffic-Jam	Car Driving
D1	Hot	Weak	Long	No
D2	Hot	Strong	Long	No
D3	Hot	Weak	Long	Yes
D4	Mild	Weak	Long	Yes
D5	Cool	Weak	Short	Yes
D6	Cool	Strong	Short	No
D7	Cool	Strong	Short	Yes
D8	Mild	Weak	Long	No
D9	Cool	Weak	Short	Yes
D10	Mild	Weak	Short	Yes
D11	Mild	Strong	Short	Yes
D12	Mild	Strong	Long	Yes
D13	Hot	Weak	Short	Yes
D14	Mild	Strong	Long	No

Sample Data Set *Auto Driving*

Day	Tem	MF (h)	MF (m)	MF(c)	Wind	MF (w)	MF (st)	Traffic	MF (l)	MF (sh)	C
D1	32	0.7	0.6	0	3	1	0	7.5	0.25	0.25	N
D2	33	0.8	0.4	0	4.5	0.25	0.3	6.8	0.18	0.37	N
D3	30	0.5	1	0	2.5	1	0	8.3	0.33	0.12	Y
D4	24	0	1	0	1.5	1	0	9	0.4	0	Y
D5	3	0	0	1	2.5	1	0	3.8	0	0.87	Y
D6	1	0	0	1	5	0	0.4	4.2	0	0.8	N
D7	8	0	0.2	1	4	0.5	0.2	2.7	0	1	Y
D8	12	0	0.47	1	3	1	0	6.7	0.17	0.38	N
D9	-5	0	0	1	2	1	0	3.5	0	0.92	Y
D10	12	0	0.47	1	2.5	1	0	4.1	0	0.82	Y
D11	15	0	0.67	0	6	0	0.5	2.3	0	1	Y
D12	22	0	1	0	5	0	0.4	7.3	0.23	0.28	Y
D13	32	0.7	0.6	0	2.5	1	0	2.6	0	1	Y
D14	25	0	1	0	4	0.25	0.3	10.3	0.53	0	N
Sum		2.7	7.41	6		9	2.1		2.09	7.81	

Fuzzy representation of the Sample Set without condition *well-defined Sample Space*

Day	MF(h)	MF(m)	MF(c)	Sum	MF(w)	MF(st)	Sum	MF(l)	MF(sh)	Sum	C
D1	0.7	0.3	0	1	1	0	1	0.75	0.25	1	N
D2	0.8	0.2	0	1	0.25	0.75	1	0.633	0.367	1	N
D3	0.5	0.5	0	1	1	0	1	0.883	0.117	1	Y
D4	0	1	0	1	1	0	1	1	0	1	Y
D5	0	0.2	0.8	1	1	0	1	0.133	0.867	1	Y
D6	0	1/15	14/15	1	0	1	1	0.2	0.8	1	N
D7	0	8/15	7/15	1	0.5	0.5	1	0	1	1	Y
D8	0	0.8	0.2	1	1	0	1	0.617	0.383	1	N
D9	0	0	1	1	1	0	1	0.083	0.917	1	Y
D10	0	0.8	0.2	1	1	0	1	0.183	0.817	1	Y
D11	0	1	0	1	0	1	1	0	1	1	Y
D12	0	1	0	1	0	1	1	0.717	0.283	1	Y
D13	0.7	0.3	0	1	1	0	1	0	1	1	Y
D14	0	1	0	1	0.5	0.5	1	1	0	1	N
Sum	2.7	7.7	3.6	1	9.25	4.75	1	6.199	7.801	1	
Ave	0.193	0.550	0.257		0.661	0.339		0.443	0.557		

Fuzzy representation of the Sample Set

- Attribute *Temperature*:

$$\mu_c(x) = \begin{cases} 1 & x < 0 \\ 1 - x/15 & 0 \leq x \leq 15 \\ 0 & x > 15 \end{cases}$$

$$\mu_m(x) = \begin{cases} 0 & x < 0 \\ x/15 & 0 \leq x < 15 \\ 1 & 15 \leq x < 25 \\ -x/10 + 3.5 & 25 \leq x < 35 \\ 0 & x > 35 \end{cases}$$

$$\mu_h(x) = \begin{cases} 0 & x < 25 \\ x/10 - 2.5 & 25 \leq x \leq 35 \\ 1 & x > 35 \end{cases}$$

- Attribute *Wind*:

$$\mu_w(x) = \begin{cases} 1 & x < 3 \\ 2.5 - x/2 & 3 \leq x \leq 5 \\ 0 & x > 5 \end{cases}$$

$$\mu_{st}(x) = \begin{cases} 0 & x < 3 \\ x/2 - 1.5 & 3 \leq x \leq 5 \\ 1 & x > 5 \end{cases}$$

- Attribute *Traffic-Jam*:

$$\mu_{sh}(x) = \begin{cases} 1 & x < 3 \\ 1.5 - x/6 & 3 \leq x \leq 9 \\ 0 & x > 9 \end{cases}$$

$$\mu_1(x) = \begin{cases} 0 & x < 3 \\ x/6 - 0.5 & 3 \leq x \leq 9 \\ 1 & x > 9 \end{cases}$$

Appendix C

Run information of WEKA.

Termination parameter is 5 (0.1)

```

==== Run information ====
Relation:  iris
Instances: 150
Attributes: 5
           a
           b
           c
           d
           e
Test mode:  split 66% train, remainder test

==== Classifier model (full training set) ====
pruned tree
-----

d <= 0.6: C1 (34.0)
d > 0.6
| d <= 1.5: C2 (32.0/1.0)
| d > 1.5: C3 (34.0/2.0)

Number of Leaves : 3
Size of the tree : 5

==== Summary ====

Correctly Classified Instances   47      92.1569 %
Incorrectly Classified Instances  4       7.8431 %

==== Confusion Matrix ====

a b c <-- classified as
15 0 0 | a = C1
 0 17 2 | b = C2
 0 2 15 | c = C3

```

Termination parameter is 10 (0.2)

```

==== Run information ====
Relation:  iris
Instances: 150
Attributes: 5
           a
           b
           c
           d
           e
Test mode:  split 66% train, remainder test

==== Classifier model (full training set) ====

pruned tree
-----
d <= 0.6: C1 (34.0)
d > 0.6
| d <= 1.5: C2 (32.0/1.0)
| d > 1.5: C3 (34.0/2.0)

Number of Leaves : 3
Size of the tree : 5

==== Summary ====

Correctly Classified Instances   47      92.1569 %
Incorrectly Classified Instances  4      7.8431 %

==== Confusion Matrix ====

 a b c <-- classified as
15 0 0 | a = C1
 0 17 2 | b = C2
 0 2 15 | c = C3

```

Termination parameter is 20 (0.4)

```

==== Run information ====
Relation:  iris
Instances: 150
Attributes: 5
          a
          b
          c
          d
          e
Test mode:  split 66% train, remainder test

==== Classifier model (full training set) ====
pruned tree
-----

d <= 0.6: C1 (34.0)
d > 0.6
| d <= 1.5: C2 (32.0/1.0)
| d > 1.5: C3 (34.0/2.0)

Number of Leaves : 3

Size of the tree : 5

==== Summary ====

Correctly Classified Instances      48      94.118 %
Incorrectly Classified Instances    3       5.882 %

==== Confusion Matrix ====

 a b c <-- classified as
15 0 0 | a = C1
 0 17 2 | b = C2
 0 0 17 | c = C3

```