

國立交通大學

電機與控制工程研究所

碩士論文

含類別屬性相依性最大化之基因演算  
模糊 ID3 方法

Genetic Algorithm Based Fuzzy ID3 Method with  
Class-Attribute Interdependence Maximization

研究生：林克勤

指導教授：張志永

中華民國九十四年七月

含類別屬性相依性最大化之基因演算  
模糊 ID3 方法

Genetic Algorithm Based Fuzzy ID3 Method with  
Class-Attribute Interdependence Maximization

學 生：林克勤

Student : Ke-Chin Lin

指導教授：張志永

Advisor : Jyh-Yeong Chang



A Thesis

Submitted to Department of Electrical and Control Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master in

Electrical and Control Engineering

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

# 含類別屬性相依性最大化之基因演算

## 模糊 ID3 方法

學生：林克勤

指導教授：張志永博士

國立交通大學電機與控制工程研究所

### 摘要

近來，許多自動獲取知識方法一直發展，一個普遍且有效的方法，主要是對於符號屬性資料的決策樹歸納，稱為 ID3 演算法。另一個被推薦的模糊 ID3 方法，他和 ID3 方法特有的特徵有高度聯繫並且擴展到應用在包含連續數值屬性的資料集。但是模糊 ID3 演算法只能處理連續數值資料，並且通常被批評為不夠高的辨識準確性。在本篇論文中，我們提出一個產生模糊決策樹的新方法，它可以接受非連續數值、連續數值或混雜型的資料並使用基因演算法調整模糊集合。此外，我們提出類別屬性相依性最大化演算法來處理資料集中特徵之最佳分段方法。接著，我們制定一個決策樹刪減的方法，以得到更精簡的規則庫。我們利用一些著名的資料集來測試我們所提出的方法，並且以兩摺交叉評比方式的結果跟 C5.0 方法比較，實驗顯示，實際上我們的方法有較好的結果；在效能上，含類別屬性相依性最大化之基因演算模糊 ID3 比起未包含類別屬性相依性最大化有較好的準確率。

# Genetic Algorithm Based Fuzzy ID3 Method with Class-Attribute Interdependence Maximization

STUDENT: Ke-Chin Lin

ADVISOR: Dr. JYH-YEONG CHANG

Institute of Electrical and Control Engineering

National Chiao-Tung University

## ABSTRACT

Many approaches to acquire knowledge automatically have been developed recently. A popular and efficient method for decision tree induction from symbolic data is ID3 algorithm. A proposed fuzzy ID3 algorithm, which is tightly connected with characteristic features of the ID3 algorithm and is extended to apply a data set containing continuous attribute values. But fuzzy ID3 algorithm can only deal with continuous data and it is often criticized to result in poor learning accuracy.

In this thesis, we proposed a genetic algorithm based fuzzy ID3 method to construct fuzzy classification system, which can accept continuous, discrete, or mixed-mode data sets. Furthermore, we proposed CAIM algorithm to deal with the best partitions of the feature of data sets. Next, we formulated a rule pruning method to obtain a more efficient rule base. We have tested our method on some famous data sets, and the results of a two-fold cross validation are compared to those by C5.0. The experiments show that our method works better in practice. The performance of the testing accuracy by our method with CAIM algorithm is better averagely than that without CAIM algorithm.

## ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor, Dr. Jyh-Yeong Chang. Without his patient guidance and inspiration during the two years, it is impossible for me to complete the thesis. In addition, I am thankful to all my Lab members for their discussion and suggestion.

Finally, I would like to express my deepest gratitude to my family, particularly my girlfriend, Wan-Lun Li. Without their strong support, I could not go through the two years.



# Content

<b>摘要.....</b>	<b>i</b>
<b>ABSTRACT.....</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>iii</b>
<b>Chapter 1. Introduction.....</b>	<b>1</b>
1.1. Research Background.....	1
1.2. Motivation.....	5
1.3. Thesis Outline.....	6
<b>Chapter 2. Genetic Algorithm Based Fuzzy ID3 Method.....</b>	<b>8</b>
2.1. Introduction to Attributes Learning.....	8
2.2. Feature Ranking.....	9
2.3. Tree Construction.....	14
2.4. Inference of Fuzzy Decision Tree.....	17
2.5 The Optimization of FID3.....	19
2.6. Pruning the Rule Base.....	26
<b>Chapter 3. Classification with CAIM Discretization algorithm....</b>	<b>30</b>
3.1. Introduction to CAIM Discretization algorithm.....	30
3.2. Class-Attribute Interdependent Discretization.....	31
3.3. Discretization Criterion.....	33
3.4. The CAIM Algorithm.....	35

3.5. GA Based Fuzzy ID3 Method with CAIM algorithm.....38

**Chapter 4. Simulation and Experiment.....41**

4.1. Description of the Data Sets.....41

4.2. Simulation and Results.....45

**Chapter 5. Conclusion.....55**

**References.....57**



## List of Figures

Fig. 1.1. Machine learning process.....	3
Fig. 2.1. The first layer of fuzzy decision tree.....	16
Fig. 2.2. The full fuzzy decision tree of this example.....	17
Fig. 2.3. Inference of the example by fuzzy decision tree.....	19
Fig. 2.4. Flowchart of genetic algorithm.....	22
Fig. 2.5. Reproduction.....	23
Fig. 2.6. Crossover.....	24
Fig. 2.7. Mutation.....	24
Fig. 2.8(a). The membership functions of “age.”.....	25
Fig. 2.8(b). The membership functions of “income.”.....	23
Fig. 2.9. The credit of each rule.....	27
Fig. 2.10. Fuzzy decision tree after pruning.....	28
Fig. 2.11. Flowchart of genetic algorithm base fuzzy ID3 method.....	29
Fig. 3.1. The CAIM algorithm flowchart.....	37
Fig. 3.2. Our GA based fuzzy ID3 method with CAIM.....	38
Fig. 3.3. CAIM discretization result.....	39
Fig. 3.4. Generated fuzzy decision after CAIM discretization.....	40
Fig. 4.1. The partial examples of the Crude oil.....	42



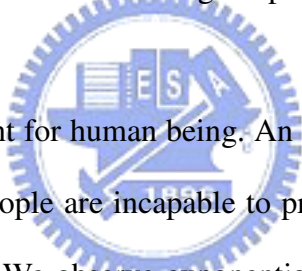
## List of Tables

Table I.	Examples of training set.....	11
Table II.	Examples with fuzzy representation of training set.....	12
Table III.	Quanta matrix for attribute $F$ and discretization scheme $D$ .....	32
Table IV.	Properties of the data sets.....	44
Table V.	Performance of the data sets before and after pruning.....	46
Table VI.	Performance of the data sets with CAIM before and after pruning.....	47
Table VII.	Comparison of the accuracy by different processing.....	48
Table VIII.	Comparison of the number of rules by different processing.....	48
Table IX.	Comparison of the testing accuracy.....	51
Table X.	Comparison of the number of the rules.....	52
Table XI.	Comparison of the best performance.....	53
Table XII.	Comparison of the accuracy by different process.....	54
Table XIII.	Comparison of the accuracy by different process (two-fold cv).....	54

# Chapter 1. Introduction

## 1.1. Research Background

Learning process is a very important element that why the natural organism or artificial system is intelligent. There are two essential kinds of learning, one is acquisition of new knowledge and the other is getting new skills. With learning, system can get experience or get some knowledge from processing. It is not enough to take down experience, the more important is that how to build program to improve their performance or adapt it at some task trough experience.



Learning is very important for human being. An infant learn how to eat and how to speak. Without learning, people are incapable to profit from their experience or to adapt to changing conditions. We observe exponential growth of the amount of data and information available on the Internet and in database systems. But the data is always disorganized and difficult to understand. Researchers often use machine learning (ML) algorithm to automate the processing and extraction of knowledge from data. Inductive ML algorithms are used to generate classification rules from class-labeled examples that are described by a set of numerical (e.g., 1, 2, 4), nominal (e.g., black, white), or continuous attributes. With analysis of the data, we can get the information or the regulations from it.

Machine learning is a burgeoning new technology with a wide range of applications. It has the potential to become one of the key components of intelligent

information systems, enabling compact generalizations, inferred from large databases of recorded information, to be applied as knowledge in various practical ways—such as being embedded in automatic processes like expert systems, or used directly for communicating with human experts and for educational purposes.

The meaning of machine learning is to develop techniques to allow computers to “learn.” Recent years, it has a great advancement on the computer capability, so it is more imperative for us to use the machine learning method. Briefly, machine learning is a method for analyzing of data sets by computer programs, it is better than the intuition of users if not possible. The purpose of system is to get knowledge form the data set, and it is often shown in the form of decision trees [1], which are the most popular choices for learning and reasoning from feature-based examples.



Machine learning has two phases, which finds the common properties between the set of examples in the database and classifies them into different classes, according to the model as shown in Fig. 1.1. In the first phase, we analyze the data set by the algorithm. We will get knowledge in the process which is in the form of decision rules or mathematical formulae. In the second phase, we use testing data to estimate the accuracy of the decision rules which generated previously. If the testing accuracy is considered acceptable, the decision rules or mathematical formulae can be built as rule-base. We can use it to classify the testing data or new data examples which the categories are not known.

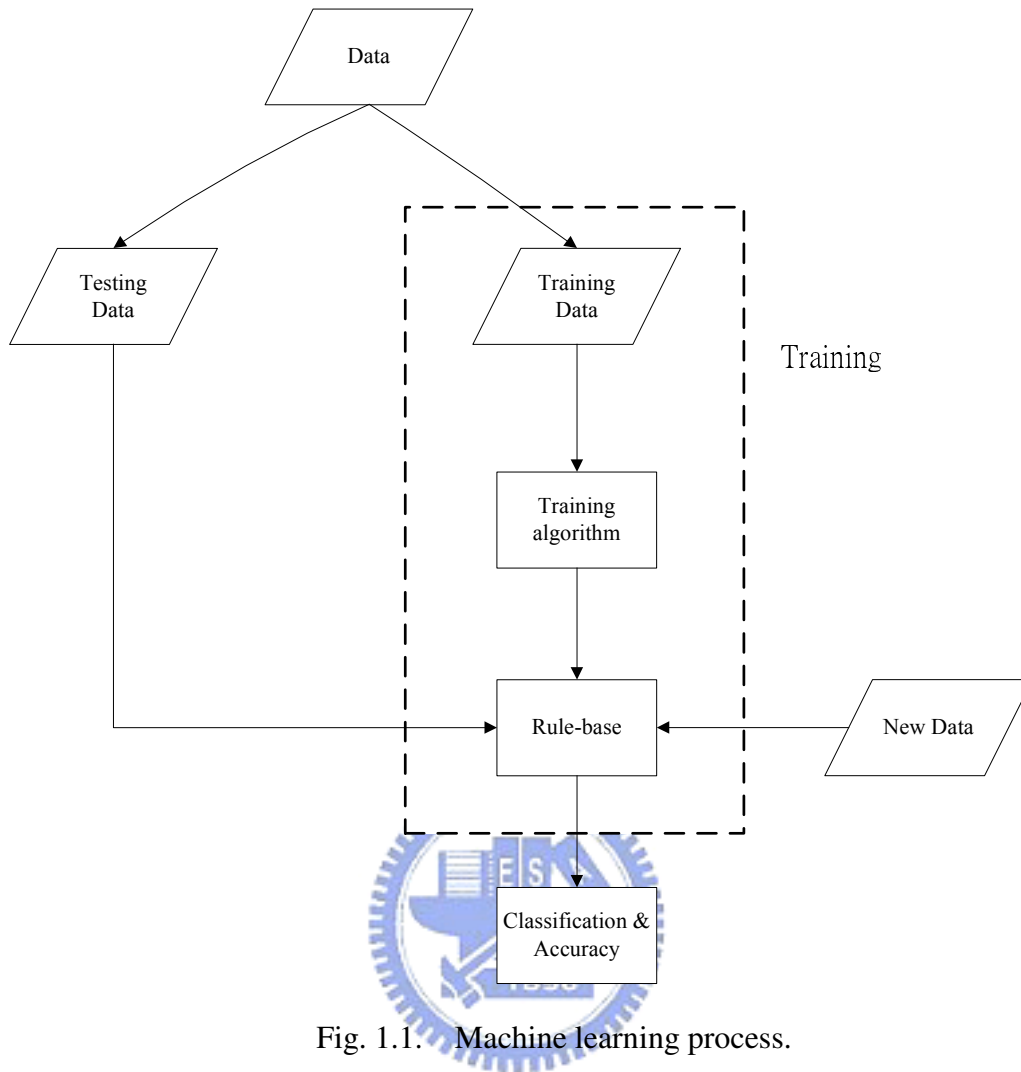


Fig. 1.1. Machine learning process.

Machine learning algorithms can be categorized in several ways. In general, they are divided into supervised and unsupervised algorithms [2]. The supervised learning algorithm is told to which class each training example belongs. In case where there is no a priori knowledge of classes, supervised learning can be still applied if the data has a natural cluster structure. Then a clustering algorithm [2] has to be run first to reveal these natural groupings. In unsupervised learning, the system learns the classes on its own. This type of learning does the classification by searching through common properties existing among the data.

There are many ways to acquire knowledge automatically. Decision tree

induction [1], has been widely used in extracting knowledge from feature-based examples for classification. A decision tree based classification method is a supervised learning method that constructs decision trees from a set of examples. The quality of a tree depends on both the classification accuracy and the size of the tree. One of the most significant developments in this domain is the ID3 algorithm, which is a popular and efficient method of making a decision tree for classification from symbolic data without much computation.

ID3 stands for “Iterative Dichotomizer (version) 3,” and is a decision tree induction algorithm, developed by Quinlan [3], and later versions including C4.5 [4] and C5.0 [5]. In the ID3 approach, we make use of the labeled examples and determine how features might be examined in sequence until all the labeled examples have been classified correctly. However, in the case of dealing with numerical data, ID3 cannot work without further modifications. If the attributes of the training set has continuous values, the algorithms must be integrated with a discretization algorithm like CART [6] and C4.5, which transforms them into several intervals. How many intervals we have to divide and the size of the decision tree is a problem to be solved because it will affect the performance of the classification. Another problem is that these decision trees are not easy to understand because we cannot know how a range of attribute is divided into intervals, and moreover most knowledge associated with human’s thinking and perception has imprecision and uncertainty. On the other hand, Umano [7] proposed Fuzzy ID3 to generate a fuzzy decision tree from fuzzy sets and applied it to diagnosis of potential transformers by analyzing gas in oil.

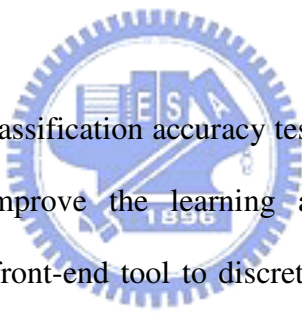
## 1.2. Motivation

Umano [7] and Janikow [8] have proposed fuzzy ID3 algorithms, which derive a modified information evaluation, however, adopt almost the same steps as what is done in traditional ID3. It is tightly connected with characteristic features of the ID3 algorithm and is extended to apply a data set containing continuous attribute values instead of symbolic attributes and generates a fuzzy decision tree using fuzzy sets defined by a user. To increase comprehensibility and avoid the misclassification due to sudden class change near the cut points of attributes, fuzzy ID3 represents attributes with linguistic variables and partitions continuous attributes into several fuzzy sets.

The construction of a fuzzy ID3 consists of three main steps: 1) generating the root node having the set of all data, 2) generating and testing new nodes to see if they are leaf nodes by some criteria, and 3) breaking the non-leaf nodes into branches by best selection of features according to feature ranking. For feature ranking, ID3 algorithm selects the feature based on the maximum information gain, which is computed by the probability of training data, but fuzzy ID3 by the degree of membership values of the training data.

Fuzzy ID3 is a typical algorithm of fuzzy decision tree induction, and from fuzzy ID3, one can extract a set of fuzzy rules, which possess many advantage such as simplicity of the rules, moderate computational effort, and easy manipulation of fuzzy reasoning. But fuzzy ID3 algorithm can only deal with continuous data and it is often criticized to result in poor learning accuracy.

In this thesis, we propose an algorithm to generate a fuzzy decision tree, which can accept continuous, discrete, or mixed-mode data sets [9], using fuzzy sets and it is tuned by genetic algorithm (GA) [10]. We improve the fuzzy ID3 algorithm in both the accuracy and the size of the tree through two key steps. First, we optimize the thresholds of leaf nodes and the mean and variance of fuzzy numbers involved by GA. Second, we prune the rules of the tree by evaluating the effectiveness of the rules, and then the reduced tree is retrained by the same GA. We can directly classify any kind of attribute included mixed-mode data by our proposed fuzzy ID3 schemes and achieves high accuracy rate due to the genetic tuning algorithm. For many famous data sets, we compare our proposed method with others to estimate the classification accuracy.



For some data sets, the classification accuracy tested by our fuzzy ID3 algorithm is not good enough. To improve the learning accuracy, we further use the discretization algorithm as a front-end tool to discretize the continuous attributes of the data sets. Here, we use the class-attribute interdependence maximization as the discretization algorithm [11] to deal with this problem. It helps us improve the performance and decrease the number of the fuzzy rules.

### **1.3. Thesis Outline**

The organization of this thesis is structured as follows. Chapter 1 introduces the role of machine learning and the motivation of this research is explained. In Chapter 2, the attribute types will be described, then we introduce genetic algorithm based fuzzy ID3 method for learning problem, and give an example to illustrate the learning process. Chapter 3 introduces the class-attribute interdependence maximization

algorithm as the front-end tool to discretize the data set for our method. For Chapter 4, the experiment of computer simulations on some famous data sets is conducted. Finally, conclusion is presented in Chapter 5.





## Chapter 2. Genetic Algorithm Based Fuzzy ID3 Method

### 2.1. Introduction to Attributes Learning

Knowledge acquisition from data is very important in knowledge engineering. The data sets are characterized by a set of attributes. There are three types of the attributes:

- 1) Continuous attributes: Continuous attributes mean that any two values of the data can be inserted with another value and it always mean the real number. In other words, continuous attributes include infinite values. For example, height and weight of human, and scores of exam are continuous attributes.
- 2) Discrete attributes: Discrete attributes are nonnumeric and are unsuitable for proximity distance based analysis. For example, a man's occupation is teacher, public servant or engineer that cannot be instead of ordinal number here.
- 3) Mix-mode attributes: The attributes include both continuous attributes and discrete attributes.

A popular and efficient method is ID3 algorithm [3]. The ID3 approach to classification consists of a procedure for synthesizing an efficient decision tree for classifying pattern that has non-numeric feature values. Fuzzy ID3 (FID3) algorithm [7], [12] extended from ID3 to incorporate fuzzy notation. The decision tree using

fuzzy ID3 algorithm is similar to that of ID3 algorithm. Fuzzy ID3 algorithm is extended to apply to a data set containing numeric feature values instead of symbolic feature and generates a fuzzy decision tree using fuzzy sets. Our algorithm is designed to handle both continuous and discrete attributes. It combines the methods of ID3 and fuzzy ID3. In the traditional fuzzy ID3 algorithm, the fuzzy sets of all continuous attributes and the threshold values of leaf node condition are user defined. A good selection of fuzzy sets and leaf node thresholds would greatly improve the accuracy of decision tree. In this thesis, we introduce genetic algorithm (GA) [13] to find out an optimal solution of the parameters of fuzzy ID3 algorithm. But the discrete attributes are divided into crisp sets, thus they have no membership functions. When deal with discrete attributes, our method is similarly to ID3. The details are described in the following sections.

## 2.2. Feature Ranking



When we start to construct decision tree, we have to choose the order of features. The process is called the Feature Ranking problem [7], [14], [15]. We can use any arbitrary order of the features, but the order of features to construct decision tree is an important issue to be investigated. With a good feature ranking, important features will be considered in the higher levels of the tree and can construct a decision tree with high accuracy and small size. The order of features is evaluated using information gain [4] here.

The fundamental premise of information theory [16] is that the generation of information can be modeled as a probabilistic process that can be measured in a manner that agrees with intuition. In accordance with this supposition, a random event  $E$  that occurs with probability  $P(E)$  is said to contain  $-\log_2 P(E)$  units of

information. If  $P(E) = 1$  (that is, the event always occurs),  $-\log_2 P(E) = 0$  and no information is attributed to it. That is because no uncertainty is associated with the event, no information would be transferred by communicating that the event has occurred. When one of two possible equally likely events occurs, the information conveyed by any one of them is  $-\log_2(1/2)$  or 1 bit. A simple example of such a situation is flipping a coin and communicating the result.

Assume that we have a set of training data  $D$ , where each data has  $l$  attributes  $A_1, A_2, \dots, A_l$  and one classified class  $C = \{C_1, C_2, \dots, C_n\}$  and fuzzy sets  $F_{i1}, F_{i2}, \dots, F_{im}$  for the attribute  $A_i$ . We assign each example a unit membership value. Let  $D^{C_k}$  to be a fuzzy subset in  $D$  whose class is  $C_k$  and  $|D|$  is the sum of the membership values in a fuzzy set of training data  $D$ .

The information gain  $G(A_i, D)$  for the  $i$ -th attribute  $A_i$  by a fuzzy set of training data  $D$  is defined by

$$G(A_i, D) = I(D) - E(A_i, D), \quad (2.1)$$

where

$$I(D) = -\sum_{k=1}^n (p_k \cdot \log_2 p_k) \quad (2.2)$$

$$E(A_i, D) = \sum_{j=1}^m (p_{ij} \cdot I(D_{F_{ij}})), \quad (2.3)$$

$$p_k = \frac{|D^{C_k}|}{|D|}, \quad (2.4)$$

$$p_{ij} = \frac{|D_{F_{ij}}|}{\sum_{j=1}^m |D_{F_{ij}}|}. \quad (2.5)$$

$I(D)$  stands for the initial entropy for the system consisting of membership

values of  $D$  labeled examples, and  $E(A_i, D)$  means the entropy of each branch according to the feature  $A_i$ . We will select the feature with maximum information gain for constructing the decision tree at root. According to  $G(A_i, D)$  of features in decreasing order, we decide the order of features from the top to the bottom of the decision tree. The feature ranking procedure will affect the performance and size of the decision tree.

We will use a training set as example to illustrate the learning process. The training set is shown in Table I. The data set contains two continuous attributes which are “age” and “income,” and one discrete attribute called “sex.” The classified classes are “have car” and “have no car.” The fuzzy sets of the continuous attributes are defined by genetic algorithm [13] that will be described in the following section. The training set with fuzzy representation based on the membership function is shown in Table II.

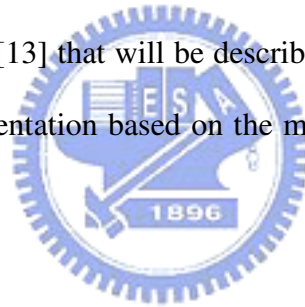


TABLE I

EXAMPLES OF TRAINING SET

ID	Class	sex	age	income
1	have car	male	27	700 K
2	have car	male	31	600 K
3	have no car	female	35	500 K
4	have no car	male	28	600 K
5	have no car	female	25	650 K
6	have car	female	33	650 K
7	have car	male	37	560 K
8	have car	female	36	580 K

TABLE II

EXAMPLES WITH FUZZY REPRESENTATION OF TRAINING SET

ID	class	sex		age		income		$\mu$
		male	female	young	old	low	high	
1	have car	1	0	0.882	0.001	0	0.134	1
2	have car	1	0	0.054	0.094	0.031	0.323	1
3	have no car	0	1	0	0.822	0.753	0	1
4	have no car	1	0	0.618	0.004	0.031	0.323	1
5	have no car	0	1	0.901	0	0	0.969	1
6	have car	0	1	0.003	0.375	0	0.969	1
7	have car	1	0	0	0.990	0.440	0.015	1
8	have car	0	1	0	0.971	0.144	0.088	1

From the theory we discuss above, we have  $|D| = 8$ ,  $|D^{C_{have\ car}}| = 5$  and  $|D^{C_{have\ no\ car}}| = 3$ , we have

$$I(D) = -\frac{5}{8} \log_2 \frac{5}{8} - \frac{3}{8} \log_2 \frac{3}{8} = 0.954.$$

For “sex,” we have

$$|D_{sex,male}| = 4, \quad |D_{sex,male}^{have\ car}| = 3, \quad |D_{sex,male}^{have\ no\ car}| = 1,$$

$$\text{and } I(D_{sex,male}) = 0.811;$$

$$|D_{sex,female}| = 4, \quad |D_{sex,female}^{have\ car}| = 2, \quad |D_{sex,female}^{have\ no\ car}| = 2,$$

$$\text{and } I(D_{sex,frmale}) = 1.$$

Now we can calculate the entropy of the branch “sex” as

$$E(sex, D) = \frac{4}{8} \times 0.811 + \frac{4}{8} \times 1 = 0.906.$$

For “age,” we have

$$|D_{age,young}| = 2.458, |D_{age,young}^{have\ car}| = 0.939, |D_{age,young}^{have\ no\ car}| = 1.519,$$

$$\text{and } I(D_{age,young}) = 0.960$$

$$|D_{age,old}| = 3.257, |D_{age,old}^{have\ car}| = 2.431, |D_{age,old}^{have\ no\ car}| = 0.826,$$

$$\text{and } I(D_{age,old}) = 0.817$$

We can calculate the entropy of the branch “age” as

$$\begin{aligned} E(age, D) &= \frac{2.458}{5.715} \times 0.960 + \frac{3.257}{5.715} \times 0.817 \\ &= 0.879. \end{aligned}$$

For “income,” we have

$$|D_{income,low}| = 1.399, |D_{income,low}^{have\ car}| = 0.615, |D_{income,low}^{have\ no\ car}| = 0.784,$$

$$\text{and } I(D_{income,low}) = 0.989;$$

$$|D_{income,high}| = 2.821, |D_{income,high}^{have\ car}| = 1.529, |D_{income,high}^{have\ no\ car}| = 1.292,$$

$$\text{and } I(D_{income,high}) = 0.9949;$$

$$E(income, D) = 0.995.$$

Thus we have the information gain for the attribute “sex” as

$$\begin{aligned} G(sex, D) &= I(D) - E(sex, D) \\ &= 0.954 - 0.906 \\ &= 0.048. \end{aligned}$$

By the same method for “age” and “income,” we have

$$G(age, D) = 0.075, G(income, D) = -0.041.$$

We will decide the order of features from the top to the bottom of the decision tree according to  $G(A_i, D)$  of features in decreasing order. Then the order of features is {age, sex, income}.

### 2.3. Tree Construction

Assume that we have a set of training data  $D$ , where each data has  $l$  continuous attributes  $A_1, A_2, \dots, A_l$  and one classified class  $C = \{C_1, C_2, \dots, C_n\}$  and fuzzy sets  $F_{i1}, F_{i2}, \dots, F_{im}$  for the attribute  $A_i$ . We assign each example a unit membership value. Let  $D^{C_k}$  be a fuzzy subset in  $D$  whose class is  $C_k$  and  $|D|$  is the sum of the membership values in a fuzzy set of training data  $D$ . An algorithm to generate a fuzzy decision tree [1], [8] is shown in the following:

1) Generate the root node and that has a set of all data, i.e., a fuzzy set of all data point with the unit membership value.

2) If a node  $t$  with a fuzzy set of data  $D$  satisfies the following conditions:

2.1) the proportion of a data set of a class  $C_k$  is greater than or equal to a threshold  $\theta_r$ , that is,

$$\frac{|D^{C_k}|}{|D|} \geq \theta_r, \quad (2.6)$$

2.2) the number of a data set is less than a threshold  $\theta_n$ , that is,

$$|D| < \theta_n, \quad (2.7)$$

2.3) there are no attributes for more classifications, then it is a leaf node,

and we record the certainties  $\frac{|D^{C_k}|}{|D|}$  with all classes at the node.

3) If it does not satisfy the above conditions, it is not a leaf node, and the branch node is generated as follows:

3.1) Divide  $D$  into fuzzy subsets  $D_1, D_2, \dots, D_m$  according to the feature  $A_i$  which has next large  $G(A_i, D)$  that will generate son

nodes. The membership value of example in  $D_j$  is the product of the membership value in  $D$  and the value of  $F_{ij}$  of the value of  $A_i$  in  $D$ .

3.2) Generate new nodes  $t_1, t_2, \dots, t_m$  for fuzzy subsets  $D_1, D_2, \dots, D_m$  and label the fuzzy sets  $F_{ij}$  to edges that connect between the nodes  $t_j$  and  $t$ .

3.3) Select the next feature for generating the son nodes by the result of feature ranking.

3.4) Replace  $D$  by  $D_j$  and repeat from step 2) recursively until the end of all paths are leaf nodes.

Now, we make the first layer of decision tree with the attribute “age” as shown in Fig. 2.1. Note that we assign each example a unit membership value first. There are two branches “young” and “old” from the root. We continue the construction process to produce the full fuzzy decision tree with other attributes until it satisfies the leaf node criteria above. For this training data, the fuzzy decision tree is shown in Fig. 2.2.



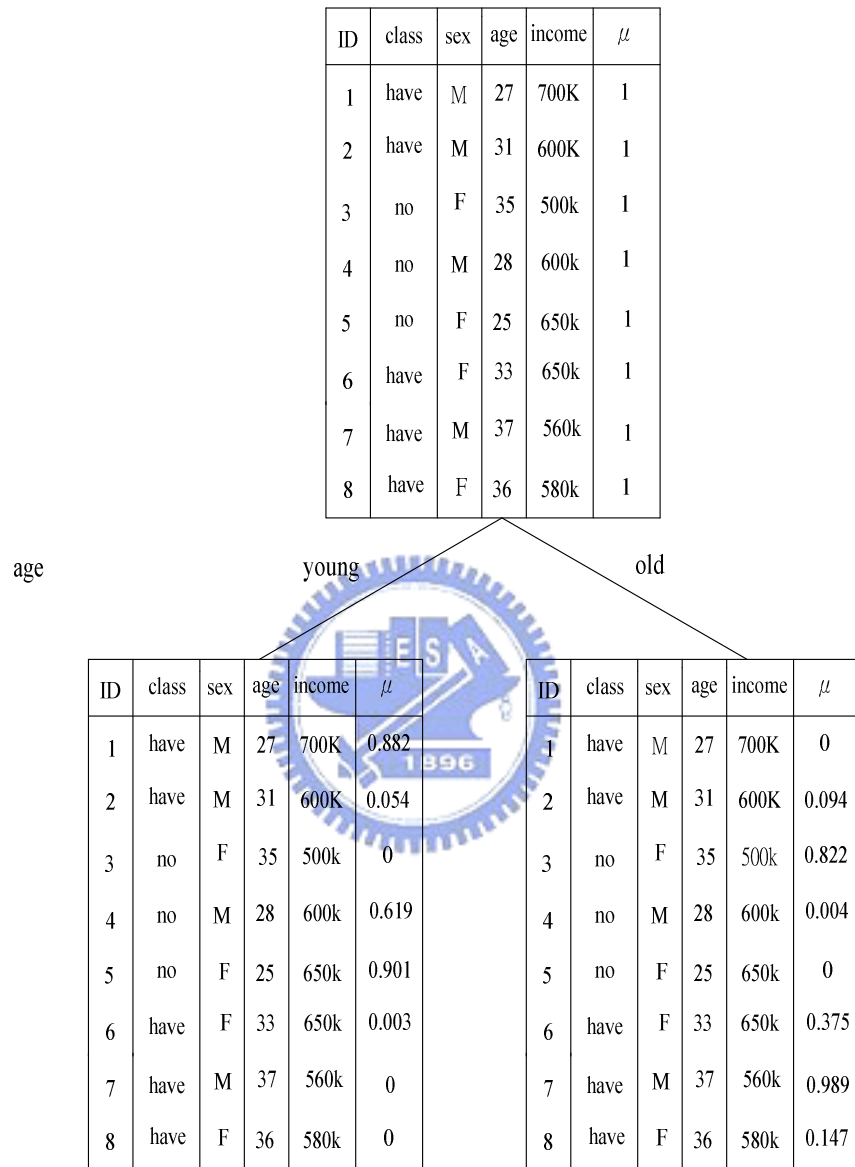


Fig. 2.1. The first layer of fuzzy decision tree.

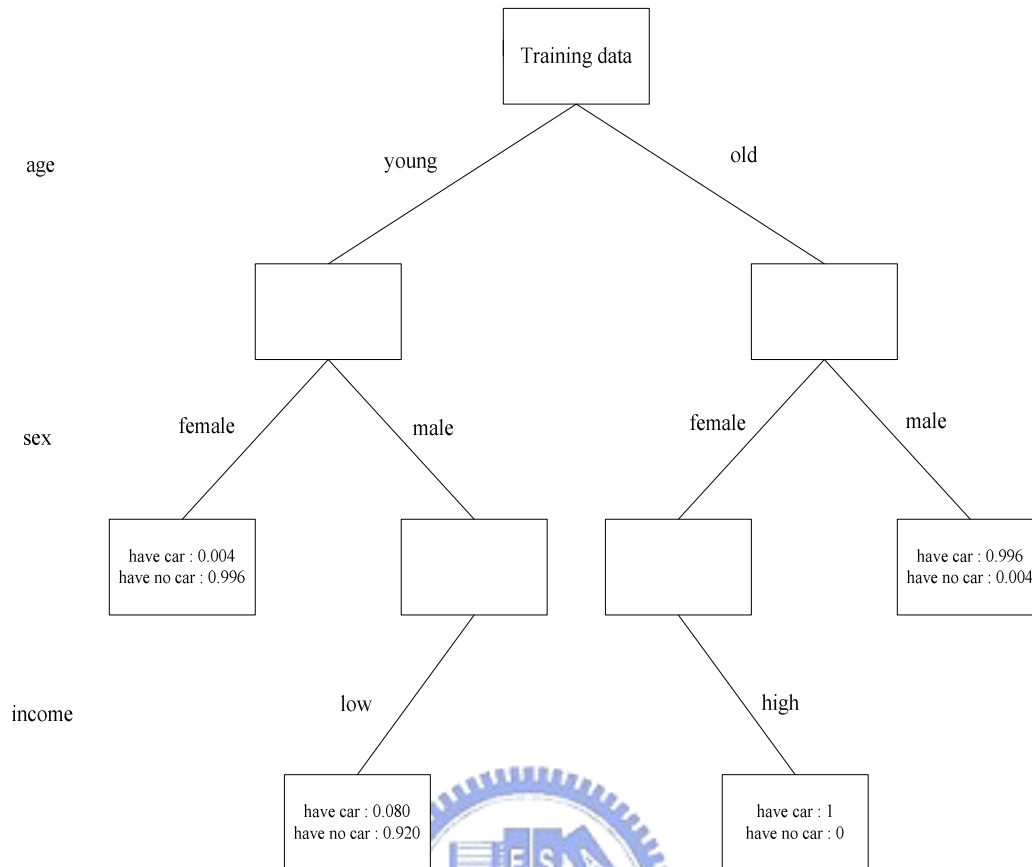


Fig. 2.2. The full fuzzy decision tree of this example.

## 2.4. Inference of Fuzzy Decision Tree

From the fuzzy decision tree we get from the training data  $D$ , we need a method to test the classification of training examples or to predict the classification of other examples. Note that we have recorded the certainties  $\frac{|D^C_k|}{|D|}$  of each class at leaf nodes as mentioned above, and it means the certainty of each class of the corresponding rule. The rule produced by each leaf node which can classify the data point to each class with the certainty value. Then the inference by fuzzy decision tree can be converted into a set of fuzzy rules. For example, the fuzzy rule extracted from the leaf node can be described as

**IF** *age is young* **AND** *sex is female*

**THEN** *have car* with certainty 0.004 and *have no car* with certainty 0.996.

For the fuzzy decision tree, there are one or more membership values between root and each leaf node because a continuous attribute value has a membership value according to the corresponding membership function. Assume that the fuzzy decision tree contains  $r$  leaf nodes, and  $n$  decision class. The steps to classify a data using obtained fuzzy rule base are described as follows:

- 1) For each  $i$  ( $1 \leq i \leq r$ ), the certainty of class  $j$  of the leaf node  $i$  multiplied by the membership values which are on the path  $i$ . Sum the  $r$  terms to get  $P(j)$  which is the possibility of the class  $j$ .
- 2) Repeat from step 1) for each  $j$  ( $1 \leq j \leq n$ ) such that all the  $P(j)$  have been computed.
- 3) The example  $e$  is assigned to the class which has the maximum value in step 2).

An illustration is shown in Fig. 2.3, where the 2-th example of Table I is tested by the fuzzy rule-base. Thus we can use these 4 rules to classify the 2-th example of Table I as follows:

$$P(\text{have car})$$

$$= 0.054 \times 0 \times 0.004 + 0.054 \times 1 \times 0.031 \times 0.080 + 0.094 \times 0 \times 0.323 \times 1 + 0.094 \times 1 \times 0.996$$
$$= 0.0934,$$

$$P(\text{have no car})$$

$$= 0.054 \times 0 \times 0.996 + 0.054 \times 1 \times 0.031 \times 0.920 + 0.094 \times 0 \times 0.323 \times 0 + 0.094 \times 1 \times 0.004$$
$$= 0.002.$$

The  $P(\text{have car})$  is maximum between all the  $P(j)$ , we assign the 2-th example to the class “have car.” Note that each rule has influence on the testing, so we use all rules to classify an example but not just depend on a single rule.

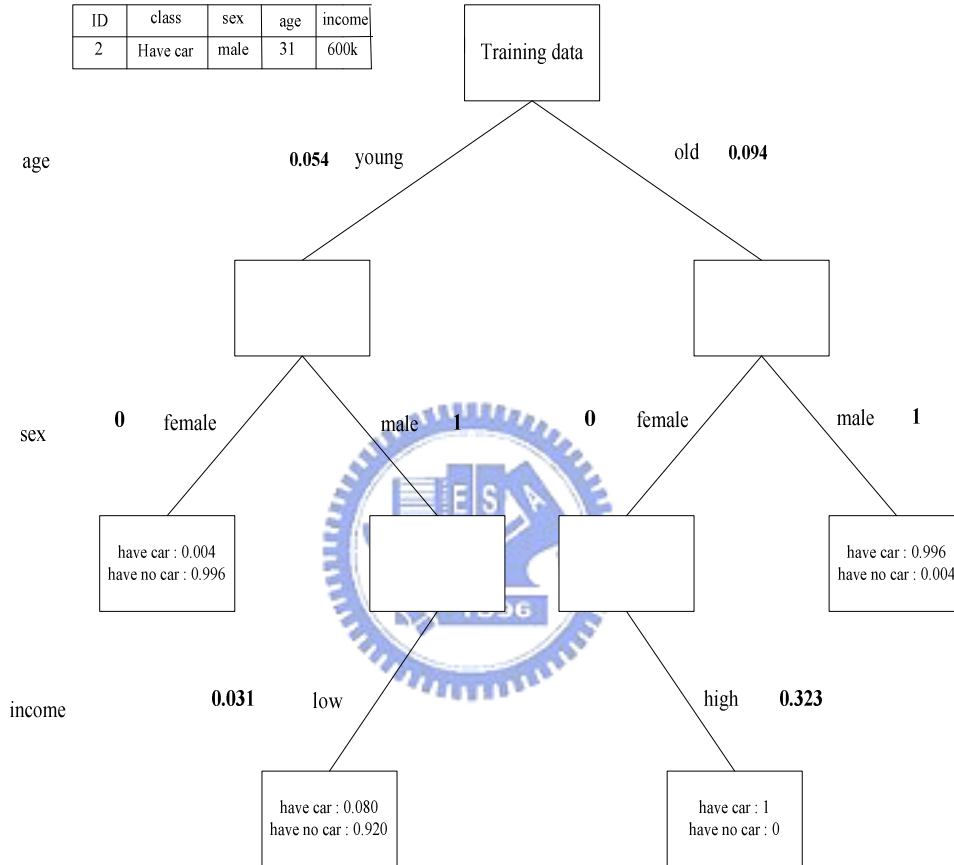


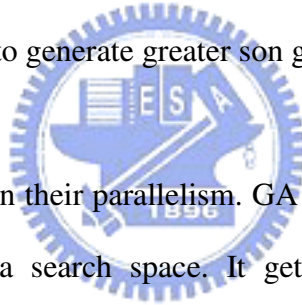
Fig. 2.3. Inference of the example by fuzzy decision tree.

## 2.5. The Optimization of FID3

From the description above, the structure of FID3 scheme is determined by the thresholds  $\theta_r$ ,  $\theta_n$ , and the membership functions of all the continuous features. A good selection of fuzzy rule base,  $\theta_r$ ,  $\theta_n$ , and the membership functions are best matched to the database to be processed, would greatly improve the accuracy of the

decision tree. To this end, any optimization algorithms seem appropriate for this purpose. In particular, genetic algorithm (GA) based scheme is highly recommended since a gradient computation for conventional optimization approach is usually not feasible for a decision tree. This is because condition-based decision path is nonlinear in nature, and hence its gradient is not defined. With this concept in mind, we will introduce in this section, genetic algorithm to search best  $\theta_r$ ,  $\theta_n$ , and the membership functions of all the continuous features for the design of fuzzy ID3.

GA is an optimization search mechanism based natural selection process. Its essential mind is to imitate the criterion “survival of the fittest” of the biology. It can choose the last generation which has the better property of the species to exchange bit information mutually to hope to generate greater son generation.



The advantage of GA is in their parallelism. GA is considering many individuals instead of an individual in a search space. It gets the global optimum rapidly, furthermore avoids the chance to fall into the local optimum.

There are several encoding of GA which depends on the problem heavily. The most common one is binary encoding which manipulate strings of binary digits (1s and 0s) called chromosomes. In this thesis, we use 6-bits to represent a parameter. We use GA to tune the thresholds  $\theta_r$ ,  $\theta_n$ , and the parameters of the membership functions of feature values. The membership function of each sub-attribute is assumed to be Gaussian-type and is given by

$$m(x) = \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad (2.8)$$

where  $x$  is the corresponding feature value of the data point with mean  $\mu$  and

standard deviation  $\sigma$ . Thus for each membership function, we have two parameters  $\mu$  and  $\sigma$  to tune. For example, assume we have a data set, which has 4 continuous attributes and 3 classes such that there are 12 membership functions. Each membership function has 2 parameters and there are 2 thresholds of leaf condition in addition. Thus we have 26 parameters to be tuned, and the length of a binary chromosome is 156. There are three operators of genetic algorithm which are reproduction, crossover, and mutation. We briefly describe how to perform these three operators. The flowchart of GA is shown in Fig. 2.4.



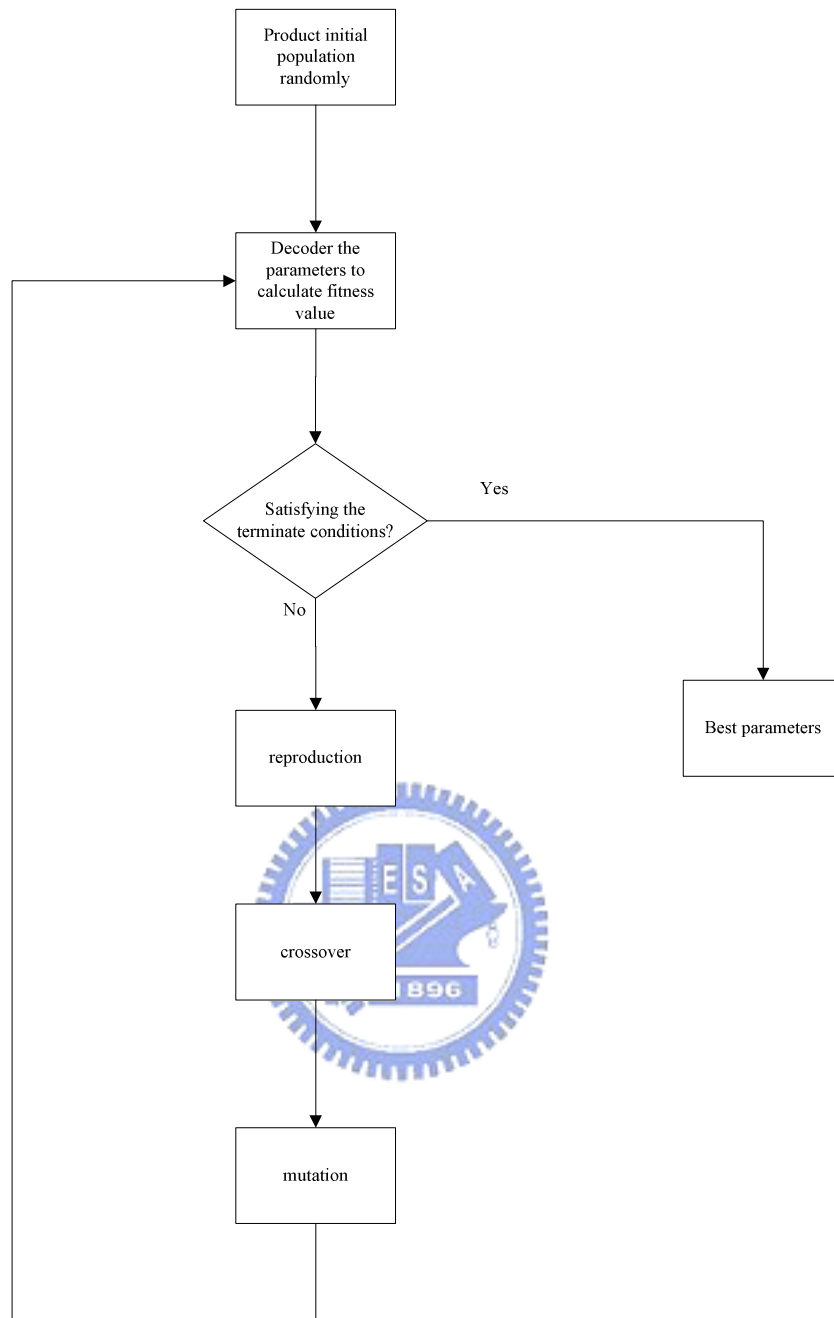


Fig. 2.4. Flowchart of genetic algorithm.

Reproduction is a process according to the fitness degree of each individual to decide which will be eliminated or copied at next generation, the individual with higher fitness value will be copied in a large number; the individual with lower fitness value will be eliminated. The potential chromosomes of the population are copied into a mating pool depending on their fitness values. The operator is shown in Fig. 2.5.

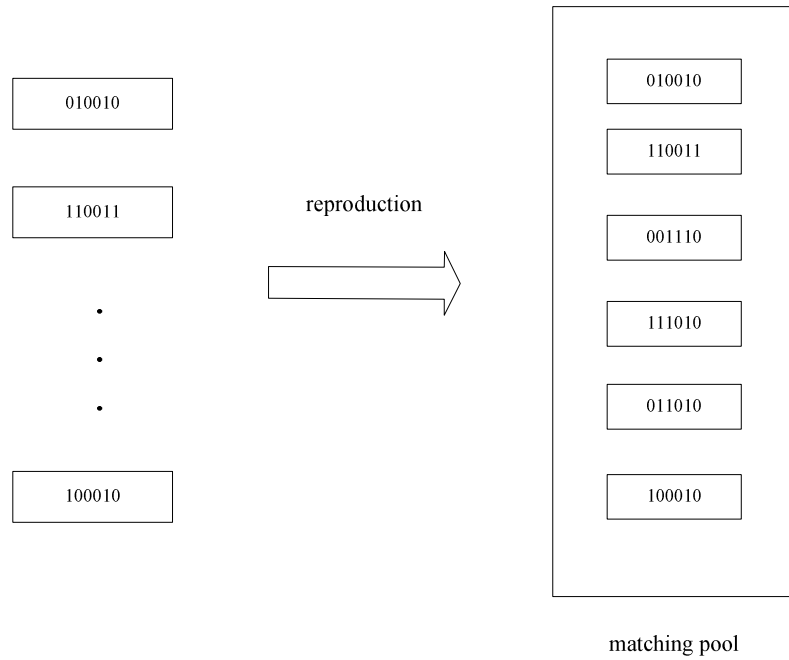


Fig. 2.5. Reproduction.

To minimize the rule number and maximize the accuracy, let the fitness function

$$f = 100(100(A_i - A_{worst})^2 + (R_{avg}/R_i)), \quad (2.9)$$

where  $A_i$  is the learning accuracy of the individual  $i$ , and  $A_{worst}$  is the worst learning accuracy of all individuals.  $R_{avg}$  is the average number of the rules of all individuals, and  $R_i$  is the number of the rules of the individual  $i$ .

Crossover is a process with selecting two potential chromosomes randomly from the matching pool, and exchange bit information mutually to produce two new individuals. Roughly speaking, it hopes to generate greater filial generation by accumulating the superior bit information of parents. An example for crossover is shown as Fig. 2.6.



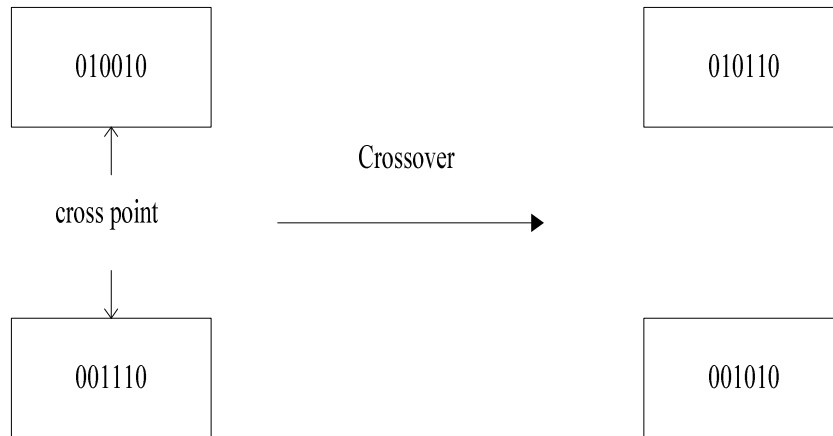


Fig. 2.6. Crossover.

Although reproduction and crossover produce many new strings, they do not introduce any new information into the population at the bit level. Mutation is introduced here, and it is the process that selects randomly string of an individual and selects randomly the mutation point to change the bit information of the string. The probability of this process is controlled by the mutation probability. For binary string, “0” is changed to “1,” and “1” is changed to “0.” The mutation is shown as Fig. 2.7.

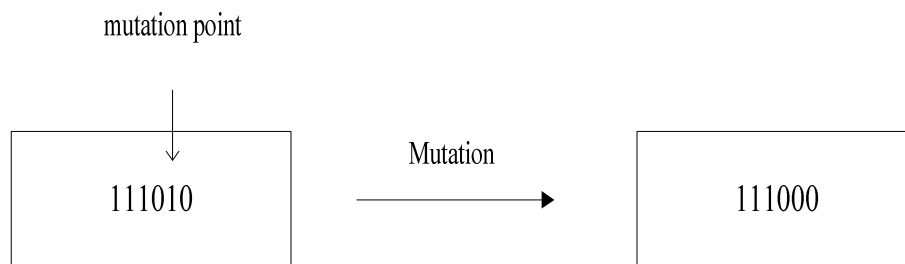
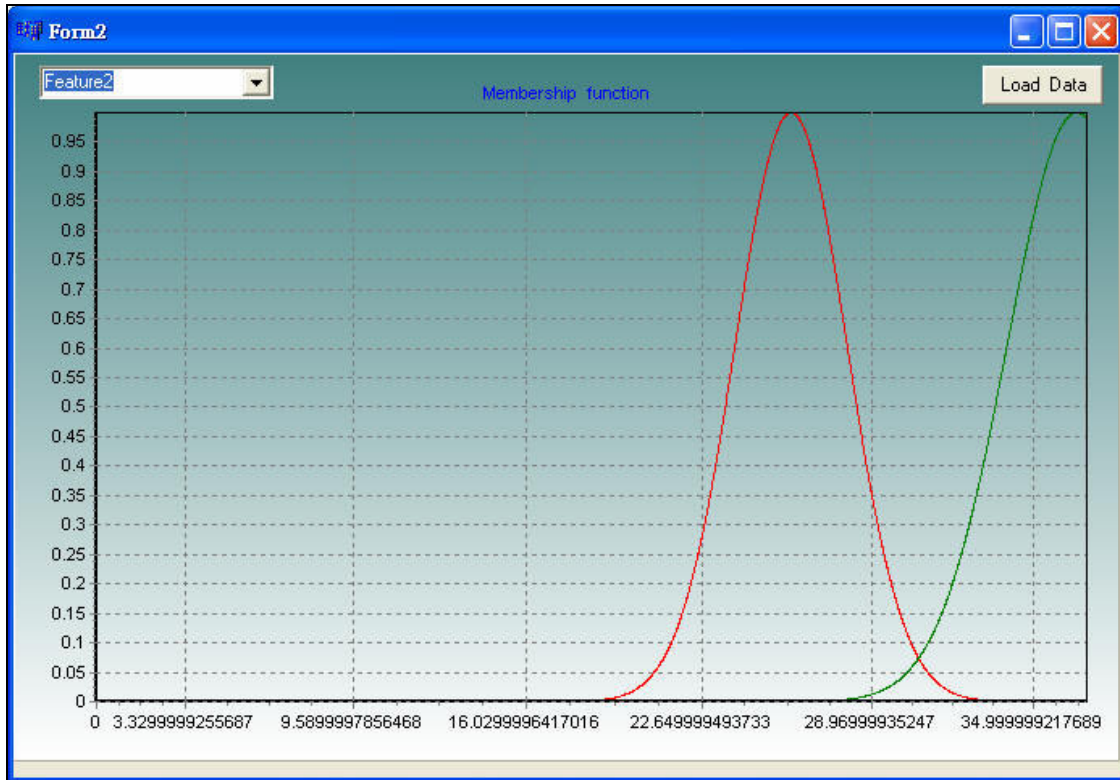
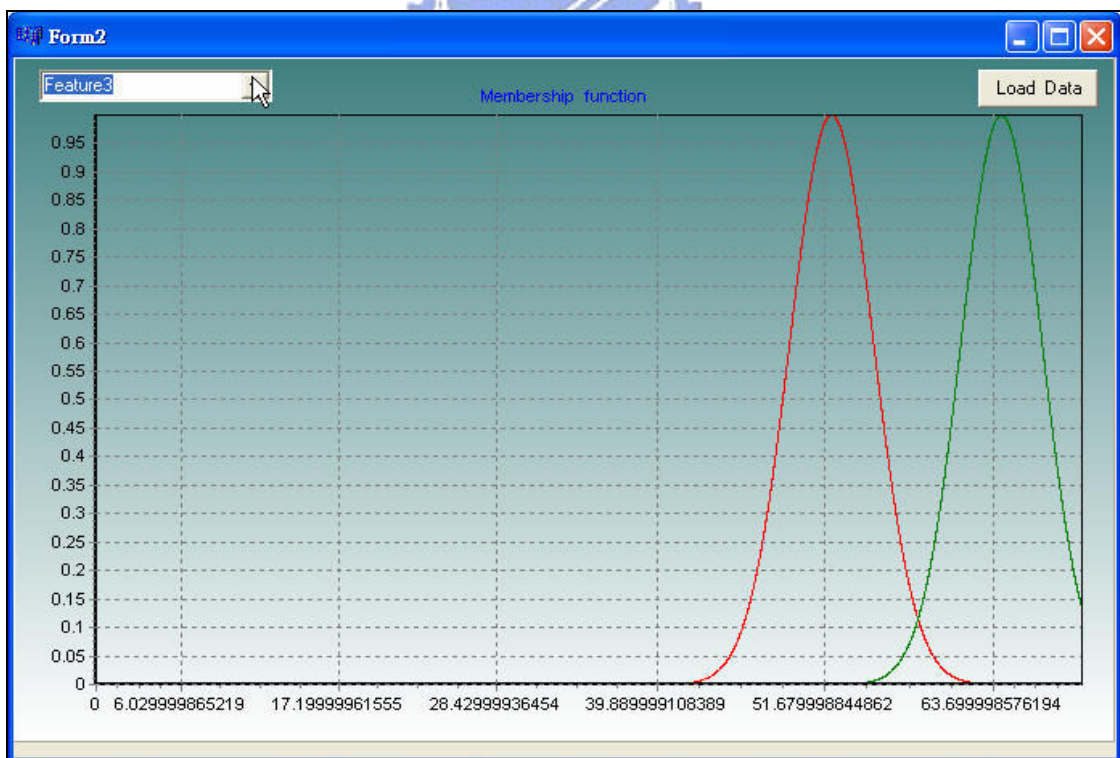


Fig. 2.7. Mutation.

After the genetic algorithm above, the system will generate two parameters of Gaussian-type membership function  $\mu$  and  $\sigma$ , and we get the parameters  $\theta_r = 0.827$ , and  $\theta_n = 0.001$  of the criteria for tree construction. The membership functions of each continuous attribute for the training set are illustrated in Fig. 2.8.



(a)



(b)

Fig. 2.8. The membership functions of each attribute for the training set. (a) The membership functions of “age.” (b) The membership functions of “income.”

## 2.6. Pruning the Rule Base

We have used the GA to improve the performance of the classification task and decrease the rule number as well. Here we propose a rule pruning method to further minimize the number of rules as follows:

- 1 ) For each rule, when any data point is classified, we maintain the production value of the membership value and the certainty of each class,  $J(n)$ .
- 2 )  $J(n)$  corresponding to the correct class of the data point gets positive sign and the others get negative sign.
- 3 ) Sum  $J(1), J(2), \dots$  for all classes of  $J(n)$ , and then we get the credit of the rule to classify this data point.
- 4 ) Repeat from 1) until all data points are classified by this rule and we get the final credit of this rule.
- 5 ) Remove the rules whose final credits are less than certain threshold and/or have big drops.

The final credit of each rule computed above represents the effectiveness of the rule in performing the classification task. If the rule is essential in classification, then it would get high credit value. On the contrary, if the credit is low, for example, less than zero, this rule could be an insignificant or redundant rule. The reason is explained as follows. The rule that classifies the data to the true class or to the wrong class will be cumulatively counted. In this way, we can prune the insignificant or inconsistent rules to obtain a smaller and efficient rule base set. After finishing rule pruning, we retune the parameters again by GA according to the pruned rule-base constructs.

For example, after we getting the credit of each rule of the training set as shown in Table I, we sort and plot the total credit of all rules as shown in Fig. 2.9. We find that the credit of the 4-th rule is much smaller than others. It suggests that the 4-th rule may be bad or redundant rule. Hence we can select a threshold between 0.449 and 0.015 and remove the redundant rule. The Pruned fuzzy decision tree of the training set as shown in Table I is shown in Fig. 2.10. The flowchart of our genetic algorithm based fuzz ID3 method is illustrated as Fig. 2.11.

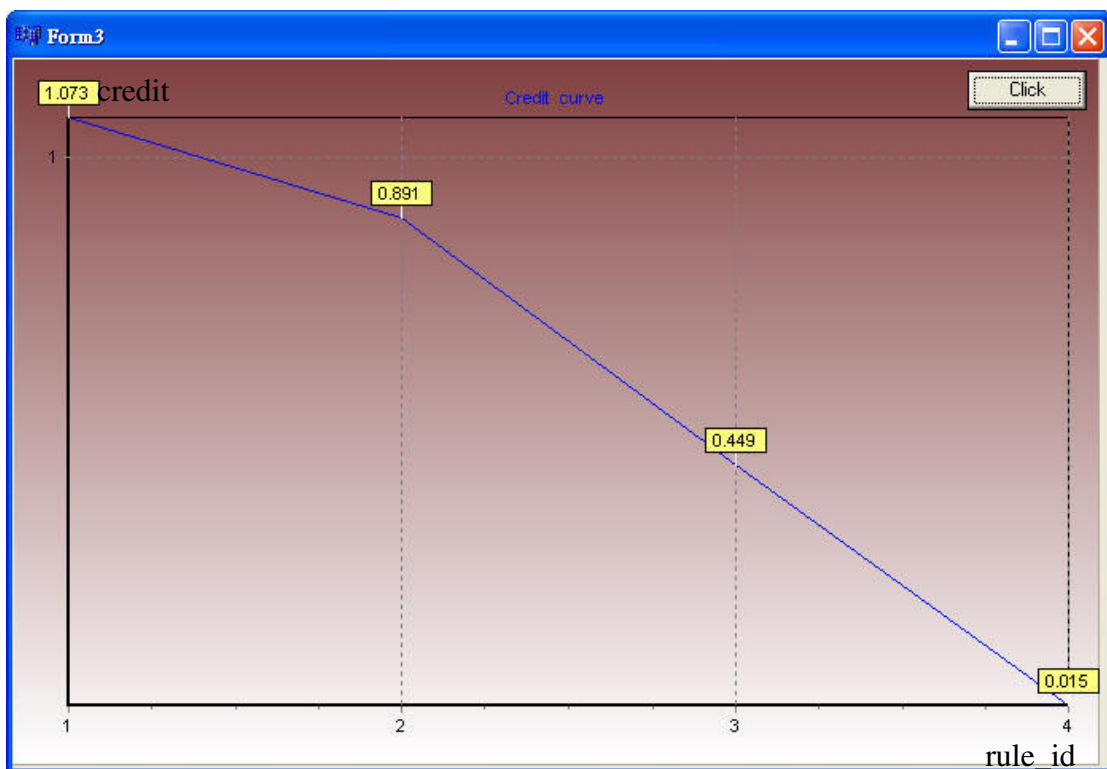


Fig. 2.9. The credit of each rule.

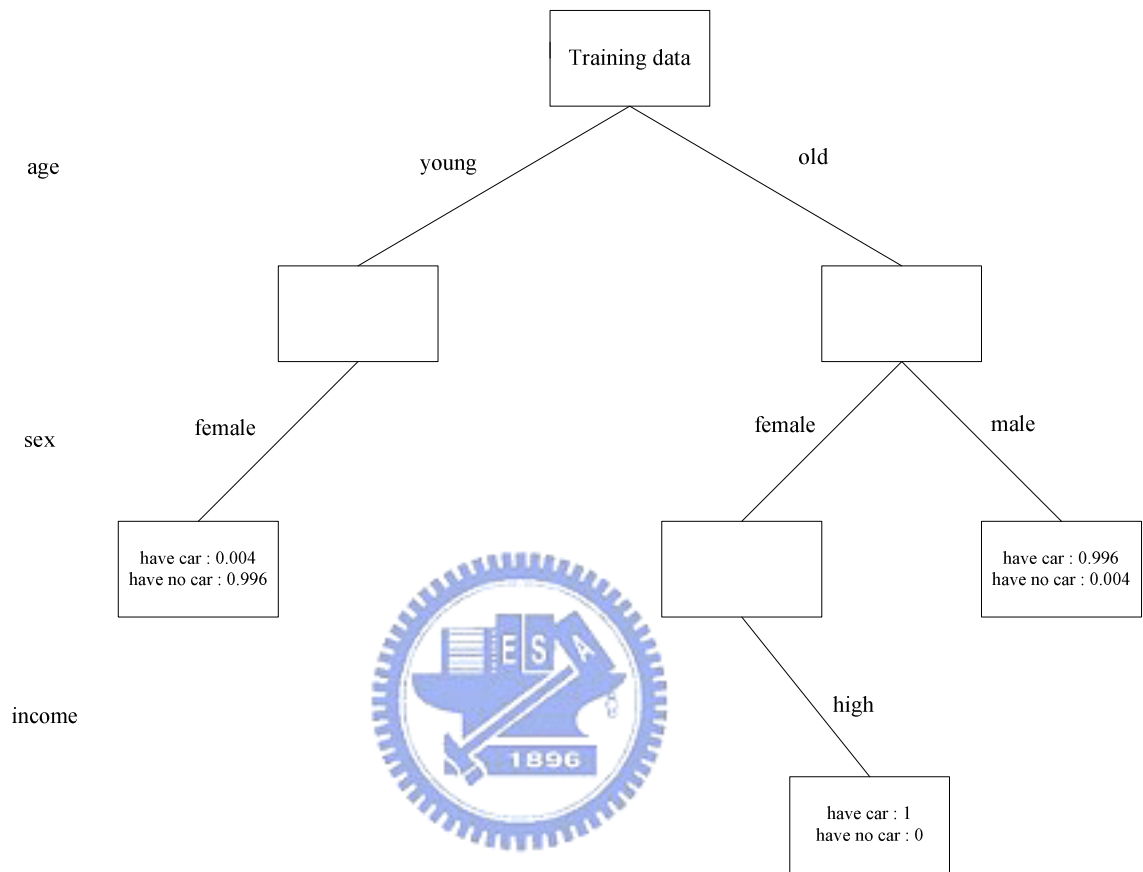


Fig. 2.10. Fuzzy decision tree after pruning.

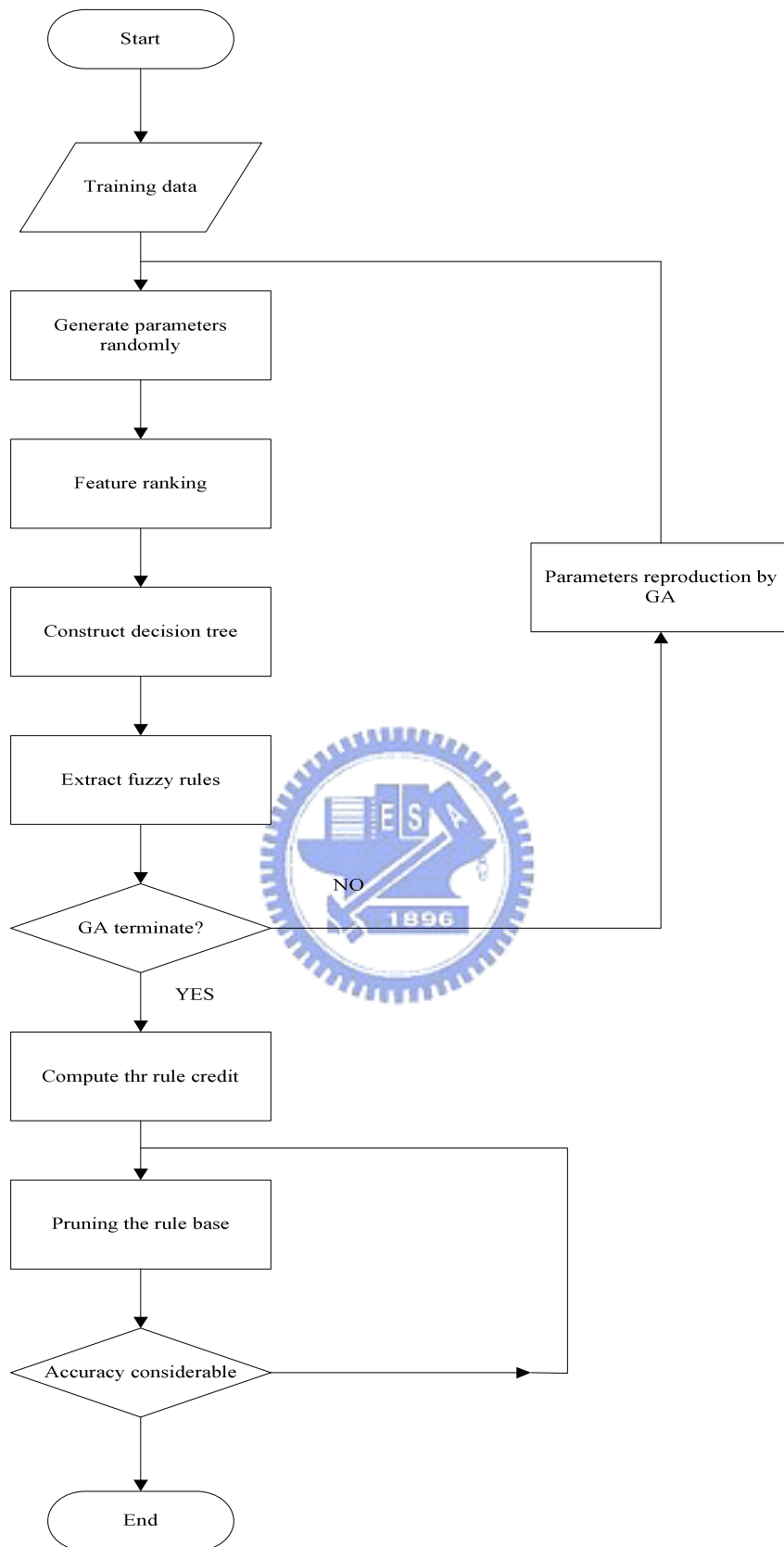
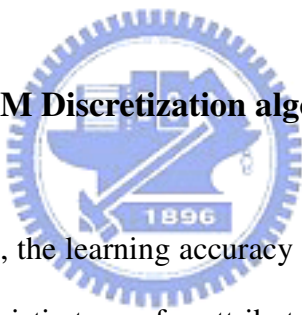


Fig. 2.11. Flowchart of genetic algorithm base fuzzy ID3 method.

# Chapter 3. Classification with CAIM Discretization algorithm

We use the class-attribute interdependence maximization (CAIM) [11] as a front-end tool for our proposed GA based fuzzy ID3 method. CAIM is a process of transforming the continuous attributes into a finite number of intervals and associating with each interval a discrete value. It helps to reduce the size of the data and improves the accuracy and the number of subsequently generated rules. First, it is instructive to explain them in detail.

## 3.1. Introduction to CAIM Discretization algorithm

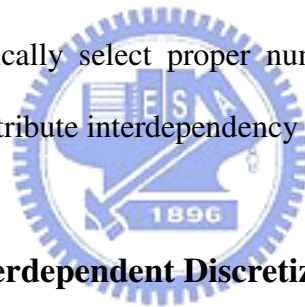


For continuous attributes, the learning accuracy of fuzzy decision tree is usually poor when the number of linguistic terms for attributes is very small. To improve the learning accuracy, we can increase the number of linguistic terms for attributes and tuning the membership functions of these terms; but it will result in the increase of the number of extracted fuzzy rules. Thus an important role to improve the performance of our method depends largely on the choice of the number of linguistic terms of continuous attributes. We can refer to the discretization algorithm, such as CAIM [11].

Discretization [17] transforms a continuous attribute's values into a finite number of intervals and associates with each interval a numerical, discrete value. For mixed-mode (continuous and discrete) data, discretization is usually performed prior to the learning process. Discretization is a two-step process. The first process is to

find the number of discrete intervals. Only a few discretization algorithms execute this automatically; on the other hand, the user must designate the number of intervals or provide a heuristic rule [10]. The second process is to find the width or the boundaries of the intervals given the range of values of a continuous attribute. Our proposed CAIM algorithm performs both tasks by automatically selecting a number of discrete intervals and finding the width of every interval based on the interdependency between classes and attribute values at the same time.

The CAIM algorithm not only discretizes an attribute into the small number of interval but also make it much easier for the subsequent machine learning task by maximizing the class-attribute interdependency. The algorithm does not require user supervision since it automatically select proper number of discrete intervals. The CAIM algorithm uses class-attribute interdependency as defined in [10].



### **3.2. Class-Attribute Interdependent Discretization**

The goal of our proposed CAIM algorithm is to find the minimum number of discrete intervals and minimum loss of the class-attribute interdependency. The algorithm uses the class-attribute interdependency information as the criterion for the optimal discretization. We introduce several basic definitions for the criterion.

For a certain classification task, assume that we have a training data set consisting of  $M$  examples, and that each example belongs to only one of the  $S$  classes.  $F$  indicates any of the continuous attributes from the mixed-mode data. Then, there exists a discretization scheme  $D$  on  $F$ , which discretizes the continuous domain of attribute  $F$  into  $n$  discrete intervals bounded by the pairs of numbers:



$$D: \{ [d_0, d_1], (d_1, d_2], \dots, (d_{n-1}, d_n] \} \quad (3.1)$$

where  $d_0$  is the minimal value and  $d_n$  is the maximum value of attribute  $F$ , and the values in (3.1) are organized in ascending order. These values constitute the boundary set  $\{d_0, d_1, d_2, \dots, d_{n-1}, d_n\}$  for the discretization  $D$ .

Each value of attribute  $F$  can be classified into only one of the  $n$  intervals defined in (3.1). With the change of discretization  $D$ , the membership value of each value within a certain interval for attribute  $F$  may also change. The class variable and the discretization variable of attribute  $F$  can be treated as two random variables, thus a two-dimensional frequency matrix (called quanta matrix) can be set up as shown in Table III.

TABLE III

QUANTA MATRIX FOR ATTRIBUTE  $F$  AND DISCRETIZATION SCHEME  $D$

Class	Interval					Class Total
	$[d_0, d_1]$	$\dots$	$(d_{r-1}, d_r]$	$\dots$	$(d_{n-1}, d_n]$	
$C_1$	$q_{11}$	$\dots$	$q_{1r}$	$\dots$	$q_{1n}$	$M_{1+}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
$C_i$	$q_{i1}$	$\dots$	$q_{ir}$	$\dots$	$q_{in}$	$M_{i+}$
$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
$C_S$	$q_{S1}$	$\dots$	$q_{Sr}$	$\dots$	$q_{Sn}$	$M_{S+}$
Interval Total	$M_{+1}$	$\dots$	$M_{+r}$	$\dots$	$M_{+n}$	$M$

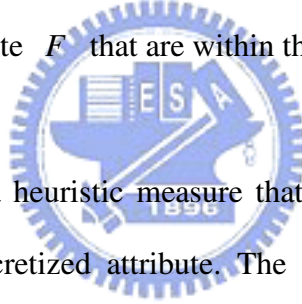
In Table III,  $q_{ir}$  is the total number of continuous values belonging to the  $i$ th class that are within interval  $(d_{r-1}, d_r]$ .  $M_{i+}$  is the total number of object belonging to the  $i$ -th class and  $M_{+r}$  is the total number of continuous values of attribute  $F$  that are within the interval  $(d_{r-1}, d_r]$ , for  $i = 1, 2, \dots, S$  and  $r = 1, 2, \dots, n$ .

### 3.3. Discretization Criterion

Given the quanta matrix as shown in Table III, the Class-Attribute Interdependency Maximization (CAIM) criterion that measures the dependency between the class variable  $C$  and the discretization variable  $D$  for attribute is defined as:

$$CAIM(C, D | F) = \frac{\sum_{r=1}^n \frac{\max_r^2}{M_{+r}}}{n},$$

where  $n$  is the number of intervals,  $r$  iterates through all intervals, i.e.,  $r=1, 2, \dots, n$ ,  $\max_r$  is the maximum value among all  $q_{ir}$  values (maximum value within the  $r$ -th column of the quanta matrix),  $i=1, 2, \dots, S$ ,  $M_{+r}$  is the total number of continuous values of attribute  $F$  that are within the interval  $(d_{r-1}, d_r]$ .



The CAIM criterion is a heuristic measure that quantifies the interdependence between classes and the discretized attribute. The criterion is independent of the number of classes and the number of the continuous attributes. It has the following properties:

- 1) The larger the value of CAIM, the higher the correlation between the class labels and the discrete intervals. The bigger the number of values belonging to class  $C_i$  within a particular interval, the higher the interdependence between  $C_i$  and the interval. The number of values belonging to  $C_i$  within the interval is the largest, and then  $C_i$  is called the leading class within the interval. The CAIM criterion accounts for the trend of maximizing the number of values belonging to a leading class within each interval by using  $\max_r$ . The value of CAIM grows when

values of  $\max_r$  grow, which relates to the increase of the interdependence between the class labels and the discrete intervals. The highest interdependence between the class labels and the discrete intervals (and, at the same time, the highest value of CAIM) is achieved when all values within a particular interval belong to the same class for all intervals. In this case,  $\max_r = M_{+r}$ , and  $CAIM = M / n$ .

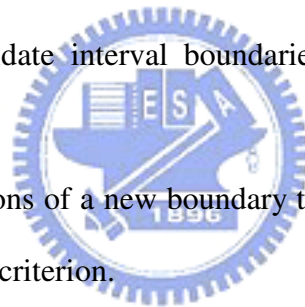
- 2) It takes on real values from the interval  $[0, M]$ , where  $M$  is the number of values of the continuous attribute  $F$ .
- 3) The squared  $\max_r$  value is divided by the  $M_{+r}$  for the reason: To eliminate the negative impact that the values belonging to classes other than the class with the maximum number of values within an interval have on the discretization scheme. The more such values the bigger the value of  $M_{+r}$  will decrease the value of CAIM.
- 4) Because the criterion favors discretization schemes with smaller number of intervals, the summed value is divided by the number of intervals  $n$ .
- 5) The  $M_{i+}$  values from the quanta matrix are not used because they are defined as the total number of objects belonging to the  $i$ -th class, which does not change with different discretization schemes.

The value of the CAIM criterion is calculated with a single pass over the quanta matrix. The CAIM criterion maximizes the class-attribute interdependency.

### 3.4. The CAIM Algorithm

The optimal discretization scheme can be found by searching over the space of all possible discretization schemes to find the one with the highest value of the CAIM criterion. Such a search for a scheme with the globally optimal value of CAIM is highly combinatorial and time consuming. Thus, the CAIM algorithm uses greedy approach, which finds local maximum values of the criterion for the approximate optimal value of the criterion. Although the method does not guarantee finding the global maximum, it is computationally efficient and effective finding the discretization scheme. The algorithm has these two tasks:

- 1) Initialize the candidate interval boundaries and the initial discretization scheme.
- 2) Constructive additions of a new boundary that results in the locally highest value of the CAIM criterion.



The pseudocode of the CAIM algorithm follows.

Given: Data consisting of  $M$  examples,  $S$  classes, and continuous attributes  $F_i$ .

For every  $F_i$  do:

Step 1.

- 1.1 Find minimum ( $d_0$ ) and maximum ( $d_n$ ) values of  $F_i$ .
- 1.2 Form a set of all distinct values of  $F_i$  in ascending order, and initialize all possible interval boundaries  $B$  with minimum, maximum and all the midpoints values of all the adjacent pairs in the set.
- 1.3 Set the initial discretization scheme as  $D : \{[d_0, d_n]\}$ , set GlobalCAIM=0.

Step 2.

- 2.1 Initialize  $k = 1$ .
- 2.2 Tentatively add an inner boundary from  $B$  which is not already in  $D$ , and calculate corresponding CAIM value.
- 2.3 After all the tentative additions have been tried, we accept the one with the highest value of CAIM.
- 2.4 If  $(\text{CAIM} > \text{GlobalCAIM} \text{ or } k < S)$  then update  $D$  with the boundary accepted in Step 2.3 and set  $\text{GlobalCAIM} = \text{CAIM}$ , else terminate.
- 2.5 Set  $k = k + 1$  and go to Step 2.2

Output: Discretization scheme  $D$

The CAIM algorithm works in a greedy top-down manner. It starts with a single interval that covers all possible values of a continuous attribute and divides it iteratively. From all possible division points that are tried (with replacement) in Step 2.2, it chooses the division boundary that gives the highest value of the CAIM criterion. The algorithm assumes that every discretized attribute needs at least number of intervals equal to the number of classes because this guarantees the discretized attribute that can improve subsequent classification. The CAIM algorithm uses trade-off between finding a discretization with the highest possible class-attribute interdependency, and a reasonable computational cost. The main advantage of this algorithm is that it finds small number of discretization intervals which gives the low computational cost, and at the same time high class-attribute interdependency. The flowchart of CAIM algorithm is shown in Fig. 3.1.

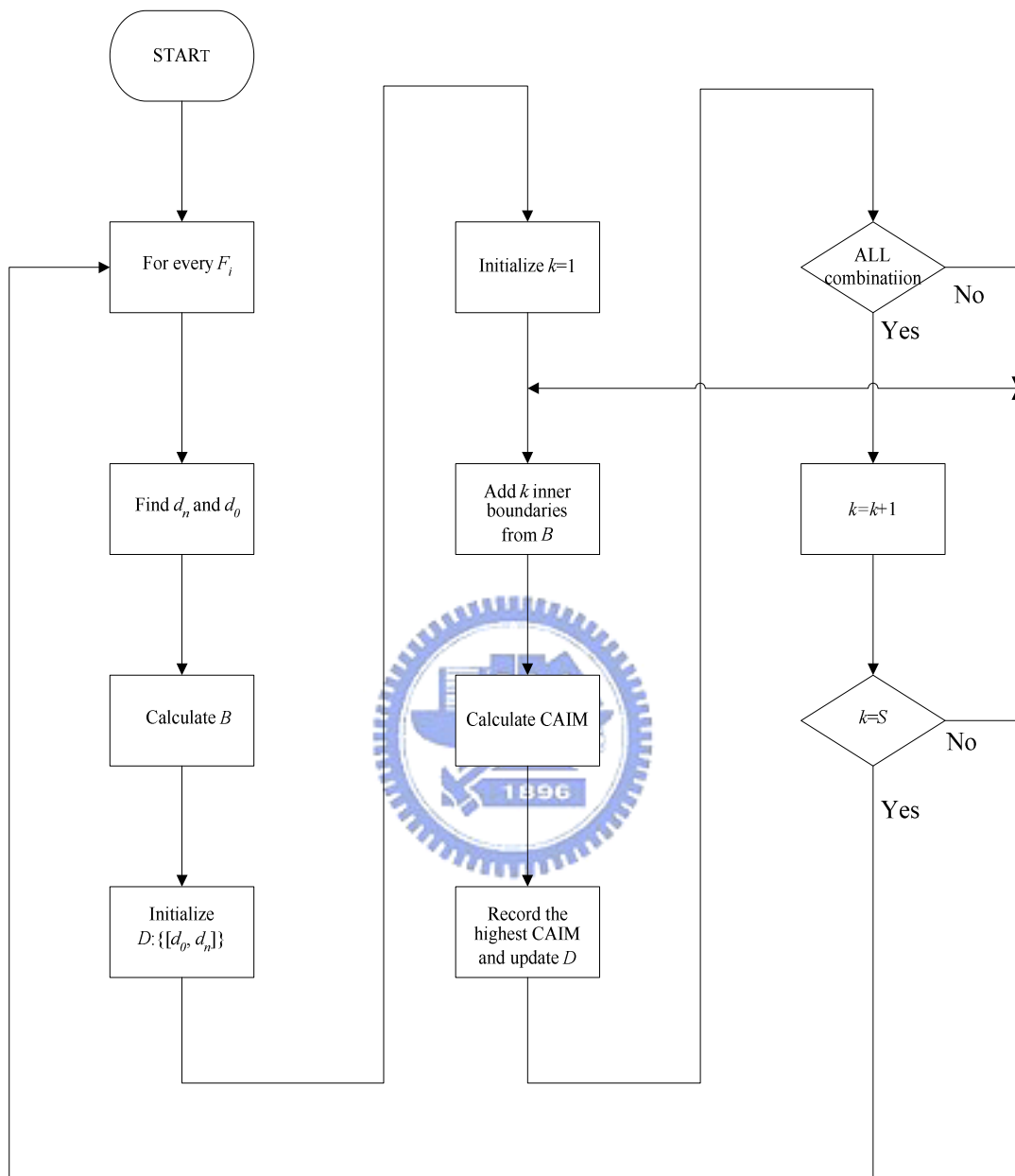


Fig. 3.1. The CAIM algorithm flowchart.

### 3.5. GA Based Fuzzy ID3 Method With CAIM Algorithm

The algorithm we proposed in Chapter 2 can accept continuous, discrete, or mixed-mode data sets. For the continuous attributes, the learning accuracy of fuzzy decision tree is sometimes poor when the number of linguistic term for attributes is very small. We use the CAIM algorithm as a front-end tool for our method. It discretizes the continuous attributes of the training data, and get the discretized data set which will replace the original training data for GA based fuzzy ID3 scheme. Fig. 3.2 gives a schematic description of our system.

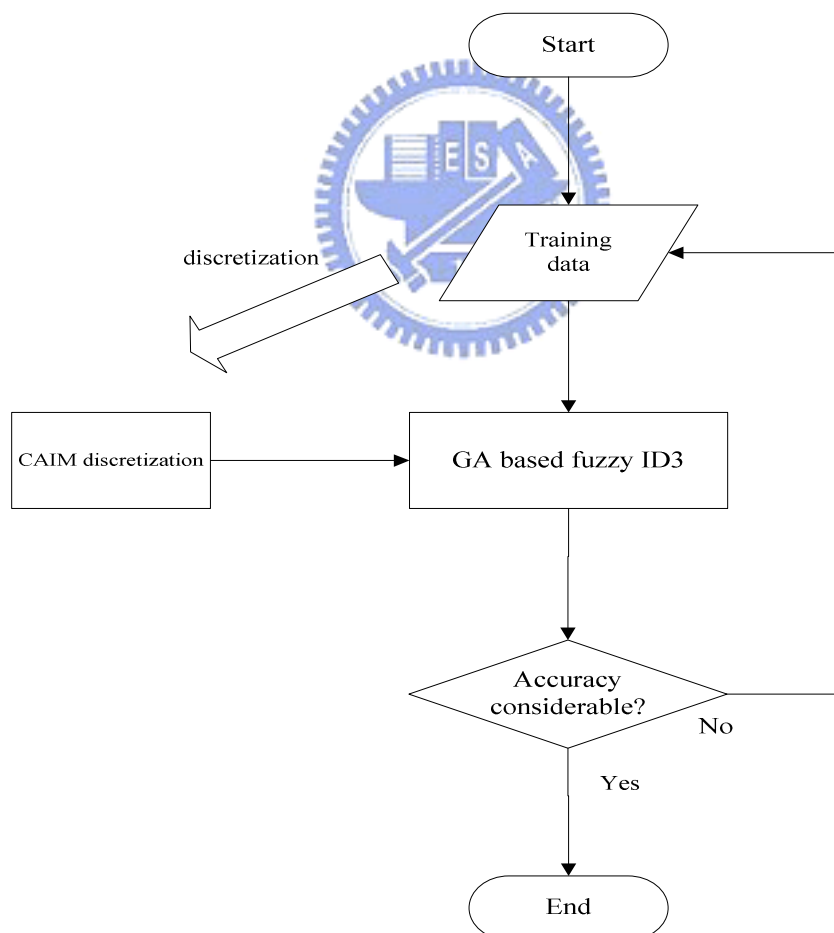


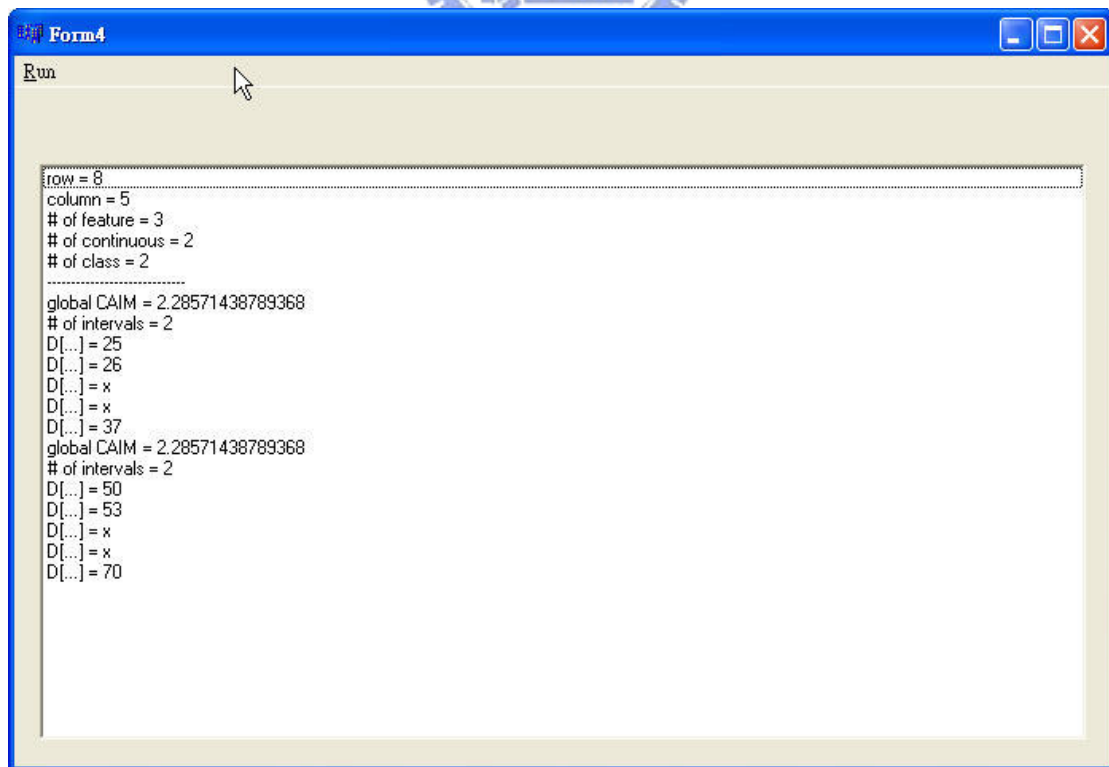
Fig. 3.2. Our GA based fuzzy ID3 method with CAIM.

We use a training set which is shown in Table I as an example to illustrate the processing. There are two continuous attributes “age” and “income” will be discretized with CAIM algorithm and two class labels in this example. For the attribute “age,” we can find that the maximum value 37 and minimum value 25. Next we form a set of all the values of “age” in ascending order, and add all the midpoints of all the adjacent pairs in the set. The CAIM algorithm will be used to get the discretization scheme. The result is shown in Fig. 3.3. We can find that the continuous attribute “age” is discretized into two discrete intervals bounded by the pairs of numbers:

$$D : \{[25,26], (26,37]\},$$

where the value 26 is the only one cut point for this scheme. By the same process for another continuous attribute “income,” we have the discretization scheme:

$$D : \{[50,53], (53,70]\}.$$



```

row = 8
column = 5
# of feature = 3
# of continuous = 2
# of class = 2
-----
global CAIM = 2.28571438789368
# of intervals = 2
D[...] = 25
D[...] = 26
D[...] = x
D[...] = x
D[...] = 37
global CAIM = 2.28571438789368
# of intervals = 2
D[...] = 50
D[...] = 53
D[...] = x
D[...] = x
D[...] = 70

```

Fig. 3.3. CAIM discretization result.



After discretization, we use the new training data for the method as mentioned in Chapter 2. With the same processing, feature ranking and tree construction, the generated decision tree is shown in Fig. 3.4. From the fuzzy decision tree, we can make the rule-base which other examples can be tested by it. The optimization and rule base pruning are also used in our proposed system.

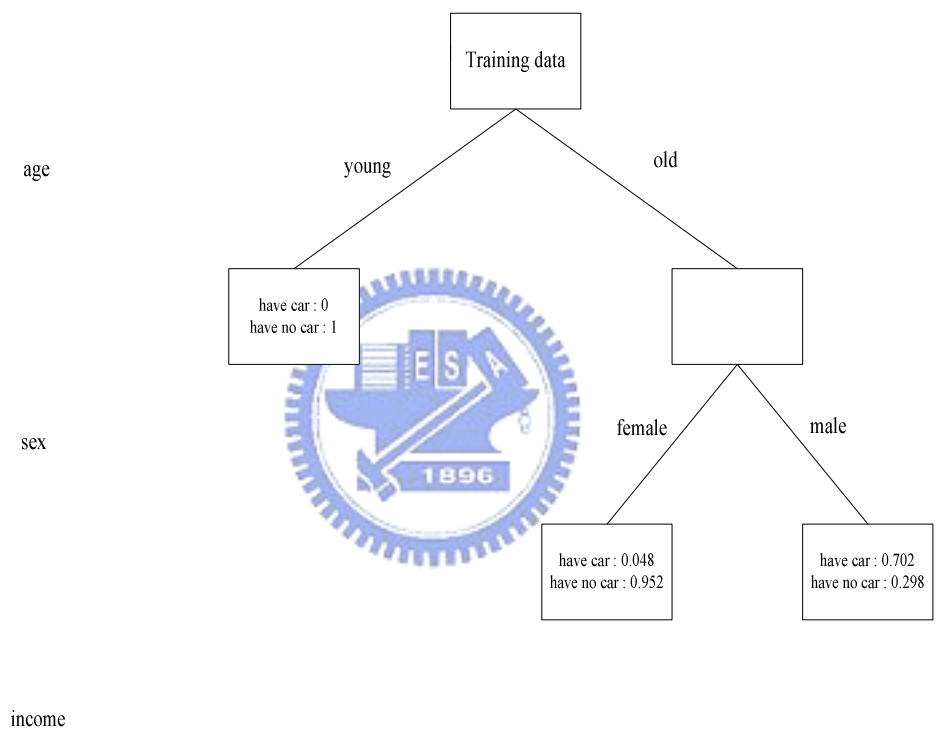



Fig. 3.4. Generated fuzzy decision after CAIM discretization.

## Chapter 4. Simulation and Experiment

As mentioned in Chapter 2, we introduce a fuzzy ID3 algorithm to construct a fuzzy classification system whose membership functions and leaf conditions are tuned by GA. In this chapter, we apply the algorithm to classify some data sets, which include continuous, discrete, and mixed-mode data sets [9], [11]. We also use this method with the class-attribute interdependence maximization algorithm to classify these data sets and compare the results. This simulation was done on Pentium 4 3.4 G personal computers.

### 4.1. Description of The Data Sets



The ten well known data sets employed for experiments are obtained from the University of California, Irvine, Repository of Machine Learning databases (UCI) [18]. We provide brief descriptions of these data sets.

- 1) **Crude\_oil**: Gerrid and Lantz analyzed Crude\_oil samples from three zones of sandstone. The Crude\_oil data set with 56 examples has five attributes and three classes named wilhelm, submuilinia, and upper. The attributes are vanadium (in percent ash), iron (in percent ash), beryllium (in percent ash), saturated hydrocarbons (in percent area), and aromatic hydrocarbons (in percent area).
- 2) **Glass Identification Database**: The data set represents the problem of identifying glass samples taken from the scene of an accident. The 214 examples were originally collected by B. German of the Home Office

Forensic Science Service at Aldermaston, Reading, Berkshire in the UK. The nine attributes are all real valued and fully known, representing refractive index and the percent weight of oxides such as silicon, sodium, and magnesium. The six classes are named as building windows float processed, building windows not float processed, vehicle windows float processed, containers, tableware, and headlamps

- 3 ) **Iris Plant Database:** The Iris data set, Fisher's classic test data (Fisher, 1936), has three classes with four-dimensional data consisting of 150 examples. The four attributes are: sepal length, sepal width, petal length, and petal width. This data set gives good results with almost all classic learning methods and has become a sort of benchmark data.
- 4 ) **Myo\_electric:** The Myo\_electric data set is extracted from a problem in discriminating between electrical signals observed at the human skin surface. This is a four-dimensional data set consisting of 72 examples divided into two classes.
- 5 ) **Norm4:** The data set has 800 examples consisting of 200 examples each from the four components of a mixture of four class 4-variate normals.
- 6 ) **BUPA liver disorders:** This UCI data set was donated by R. S. Forsyth. The problem is to predict whether or not a male patient has a liver disorder based on blood tests and alcohol consumption. There are two classes, six continuous attributes, and 345 examples.
- 7 ) **Promoter Gene Sequences Database:** Promoters have a region where a protein (RNA polymerase) must make contact and the helical DNA sequence must have a valid conformation so that the two pieces of the contact region spatially align. The data set with 106 examples has 57 attributes and two classes. All attributes are discrete.

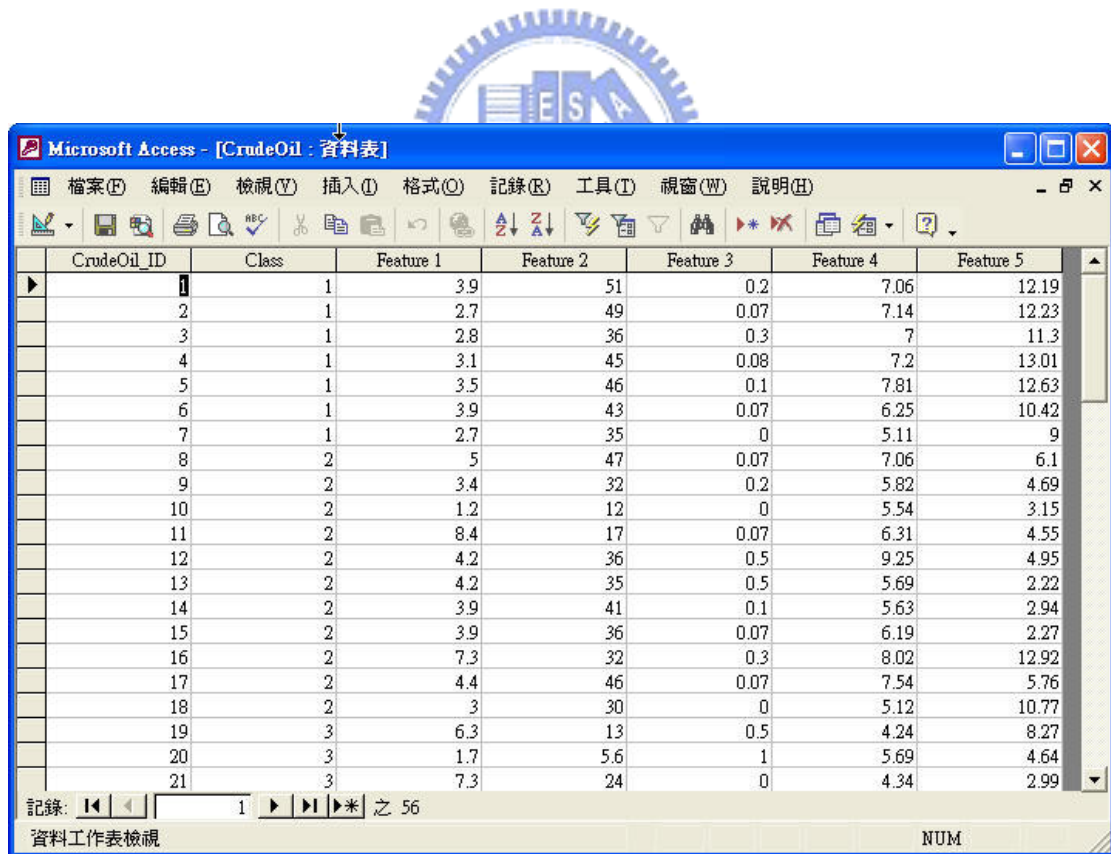
- 8) **StatLog Project Heart Disease dataset:** This UCI data set is from the Cleveland Clinic Foundation, courtesy of R. Detrano. The problem concerns the prediction of the presence or absence of heart disease given the results of various medical tests carried out on a patient. There are two classes, seven continuous attributes, six discrete attributes, and 270 examples.
- 9) **Golf:** The data set with 28 examples has four attributes and two classes named play, and don't play. There are 2 continuous and 2 discrete attributes. The attributes are outlook, temperature, humidity, and windy.
- 10) **StatLog Project Australian Credit Approval:** This credit data originates from Quinlan. This file concerns credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. The Australian data set with 690 examples has 14 attributes and two classes. There are 6 continuous and 8 discrete attributes.

These characters are described above. In order to clearly summarize the ten data sets, we list the properties of them in Table IV and the partial examples of our testing data sets are illustrated in Fig. 4.1.

TABLE IV

PROPERTIES OF THE DATA SETS

Data set	# of examples	# of attributes	# of continuous attributes	# of classes
Crude oil	56	5	5	3
Glass	214	9	9	6
Iris	150	4	4	3
Myo_electric	72	4	4	2
Norm4	800	4	4	4
Bupa	345	6	6	2
Promoters	106	57	0	2
Heart	270	13	6	2
Golf	28	4	2	2
Australian	690	14	6	2



The screenshot shows a Microsoft Access window titled "Microsoft Access - [CrudeOil : 資料表]". The table displayed has the following data:

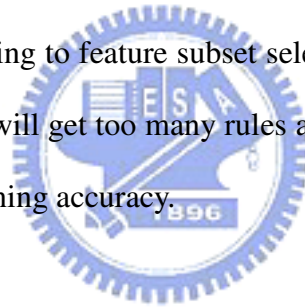
CrudeOil_ID	Class	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
1	1	3.9	51	0.2	7.06	12.19
2	1	2.7	49	0.07	7.14	12.23
3	1	2.8	36	0.3	7	11.3
4	1	3.1	45	0.08	7.2	13.01
5	1	3.5	46	0.1	7.81	12.63
6	1	3.9	43	0.07	6.25	10.42
7	1	2.7	35	0	5.11	9
8	2	5	47	0.07	7.06	6.1
9	2	3.4	32	0.2	5.82	4.69
10	2	1.2	12	0	5.54	3.15
11	2	8.4	17	0.07	6.31	4.55
12	2	4.2	36	0.5	9.25	4.95
13	2	4.2	35	0.5	5.69	2.22
14	2	3.9	41	0.1	5.63	2.94
15	2	3.9	36	0.07	6.19	2.27
16	2	7.3	32	0.3	8.02	12.92
17	2	4.4	46	0.07	7.54	5.76
18	2	3	30	0	5.12	10.77
19	3	6.3	13	0.5	4.24	8.27
20	3	1.7	5.6	1	5.69	4.64
21	3	7.3	24	0	4.34	2.99

The status bar at the bottom indicates "記錄: 1 之 56" and "資料工作表檢視".

Fig. 4.1. The partial examples of the Crude oil.

## 4.2. Simulation and Results

We use all the data sets to be the training data and the same examples to be the testing data for the performance evaluation with our proposed GA based fuzzy ID3 method. The performance includes the testing accuracy and the number of fuzzy rules. We record the accuracy and the number of fuzzy rules after testing. As mentioned in Sec. 2.6, we know it is necessary to prune the redundant rules to get an efficient rule base both in terms of size and quality, and we proposed a method to prune the rule base. On rules that can significantly reduce the learning accuracy in case we remove it. We take down the accuracy and the number of fuzzy rules before and after pruning as shown in Table V. For classifying Glass data set, we consider only five attributes that are Na, Mg, Al, K, Ba according to feature subset select [19]. If we do not reduce the attributes of this data set, we will get too many rules after tree construction, but it will not help in increasing the learning accuracy.



From Table V, we find that the data sets of Myo\_electric, Promoters, Golf, and Australian, the accuracy remain the same before and after rule pruning. For these data sets, we get less number of rules and remain the training accuracy which is the best condition of rule pruning. For the others, there is a little degradation in the accuracy and decreasing the rule numbers. This has happened possibly because the rule pruning process has removed some rules, which were correctly classifying these data sets and the residual rules are not able to correctly classify few examples. We can also see that the number of the rules is decreased for all data sets, which shows the effectiveness of our rule pruning process.

TABLE V

PERFORMANCE OF THE DATA SETS BEFORE AND AFTER PRUNING

Data set	Before rule pruning		After rule pruning	
	# of rules	Training acc.	# of rules	Training acc.
Crude_oil	9.0	100.0	7.0	98.2
Glass	55.0	77.6	23.0	76.2
Iris	8.0	99.3	6.0	98.7
Myo_electric	4.0	98.6	2.0	98.6
Norm4	35.0	96.0	23.0	95.2
Bupa	11.0	75.7	6.0	74.5
Promoters	7.0	85.8	4.0	85.8
Heart	11.0	84.7	9.0	84.1
Golf	9.0	100.0	6.0	100.0
Australian	5.0	87.0	3.0	85.6



We use CAIM algorithm as a front-end tool to discretize these ten data sets. The data sets after discretization are used to be the training data and the same examples to be the testing data for our proposed GA based fuzzy ID3. By the similar analysis, we do the performance evaluation, and record the accuracy and the number of fuzzy rules. Next, we use our rule pruning method to decrease the size of rule base. From Table IV, we know that the data set Promoters has no continuous attribute, so it is not tested in this procedure. The performance with the data sets before and after pruning is shown in Table VI.

TABLE VI

PERFORMANCE OF THE DATA SETS WITH CAIM BEFORE AND AFTER  
PRUNING

Data set	Before rule pruning		After rule pruning	
	# of rules	Training acc.	# of rules	Training acc.
Crude_oil	13.0	100.0	11.0	100.0
Glass	48.0	79.4	18.0	76.2
Iris	8.0	98.7	5.0	98.2
Myo_electric	4.0	98.6	2.0	98.6
Norm4	37.0	96.4	23.0	95.4
Bupa	17.0	77.6	15.0	76.2
Heart	10.0	84.8	8.0	84.4
Golf	9.0	100.0	7.0	100.0
Australian	4.0	86.5	3.0	85.2

From Table VI, we find that the data sets of Crude\_oil, Myo\_electric, Promoters, and Golf, the accuracy are the same before and after rule pruning. For the others, there is a little reduction in the training accuracy. It also shows the effectiveness of our proposed rule pruning method.



TABLE VII

COMPARISON OF THE ACCURACIES BY DIFFERENT PROCESSING

Data set	before pruning		After pruning	
	without CAIM	CAIM	without CAIM	CAIM
Crude_oil	<b>100.0</b>	<b>100.0</b>	98.2	<b>100.0</b>
Glass	77.6	<b>79.4</b>	<b>76.2</b>	<b>76.2</b>
Iris	<b>99.3</b>	98.7	<b>98.7</b>	98.2
Myo_electric	<b>98.6</b>	<b>98.6</b>	<b>98.6</b>	<b>98.6</b>
Norm4	96.0	<b>96.4</b>	95.2	<b>96.8</b>
Bupa	75.7	<b>77.6</b>	74.5	<b>76.2</b>
Heart	84.7	<b>84.8</b>	84.1	<b>84.4</b>
Golf	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
Australian	<b>87.0</b>	86.5	<b>85.6</b>	85.2
<b>RANK (mean)</b>	1.4	<b>1.2</b>	1.4	<b>1.2</b>

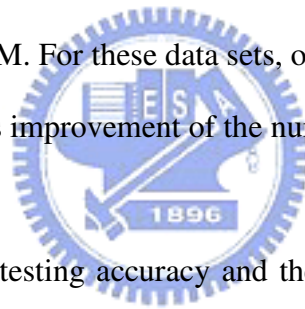


TABLE VIII

COMPARISON OF THE NUMBER OF RULES BY DIFFERENT PROCESSING

Data set	before pruning		After pruning	
	without CAIM	CAIM	without CAIM	CAIM
Crude_oil	<b>9.0</b>	13.0	<b>7.0</b>	11.0
Glass	55.0	<b>48.0</b>	23.0	<b>18.0</b>
Iris	<b>8.0</b>	<b>8.0</b>	6.0	<b>5.0</b>
Myo_electric	<b>4.0</b>	<b>4.0</b>	<b>2.0</b>	<b>2.0</b>
Norm4	<b>35.0</b>	37.0	<b>23.0</b>	26.0
Bupa	<b>11.0</b>	17.0	<b>6.0</b>	15.0
Heart	11.0	<b>10.0</b>	9.0	<b>8.0</b>
Golf	<b>9.0</b>	<b>9.0</b>	<b>6.0</b>	7.0
Australian	5.0	<b>4.0</b>	<b>3.0</b>	<b>3.0</b>
<b>RANK (mean)</b>	<b>1.3</b>	<b>1.3</b>	<b>1.3</b>	1.4

We arrange the performances which are shown in Table V and Table VI. Direct comparison of results can be seen by looking at the RANK column in Table VII that shows the accuracy. Here, we give a rank to the performance of each data set by the processing with and without CAIM. In the following tables, a bold face is emphasized to the number of the top one, i.e., winner. The last row is the RANK (mean), which is defined as the average of all the rank obtained of the data sets above. Table VIII shows the classification results in terms of number of generated rules. From Table VII, we find that the data sets discretized by CAIM algorithm have the higher RANK in both processes before and after pruning. By the comparison of these ten data sets, it shows that the CAIM algorithm significantly improves accuracy of the results. In Table VIII, the RANK of rule numbers generated with CAIM is smaller than or equal to that generated without CAIM. For these data sets, our proposed method with CAIM algorithm has not conspicuous improvement of the number of rules.



The performance of the testing accuracy and the number of generated rules by our proposed GA based fuzzy ID3 method with or without CAIM algorithm has been discussed. So far, we have not evaluated the generalization ability of the testing accuracy and the rules extracted by our scheme. Next, we compare our method with C5.0 [5]. The reason why we choose C5.0 is a decent version of C4.5 and is the state-of-the-art algorithm which works well for many decision-making problems.

We use two-fold cross validation testing which divides the each data set in two folds. The instances are randomly divided among the two folds. One of the two folds is then trained using our proposed learning algorithm and C5.0. Then the learned structure is then tested against the other fold. The same procedure is repeated considering the second fold to be the train data and the first fold to be the testing data.

Average accuracy and the number of rules are recorded. This procedure is repeated six times. Note that, C5.0 whose demonstration version is limited up to 400 examples, and free download from Rule Quest Research Data Mining Tools [5]. We use this demonstration version of C5.0 as the learning tool.

The comparison of the testing accuracy of our method and that from C5.0 is shown in Table IX. It takes down the testing accuracy from two-fold cross validation repeated six times on each data set. On average, we find that our rule-base outperforms C5.0 in eight out of ten data sets. Thus our system has better generalization ability than C5.0 and except for the data sets Glass and Bupa. The results of our method and C5.0 are also compared with respect to the average number of rules. Table X shows the comparison of the number of rules generated by these two methods at the same experiment. We find that our rule-base outperforms C5.0 in five out of ten data sets. But, the total average number of the rules on our rule-base is 7.18, which less than 8.17 of C5.0. It is evident that our approach tends to produce more concise rule sets than C5.0.

Table XI lists the maximum testing accuracy of the six for our rule-base and C5.0 in Table IX. It also shows the corresponding number of the rules in the experiment. With respect to the testing accuracy shown in Table XI, our rule-base is still superior to C5.0 in six data sets and ties one.

**TABLE IX**  
**COMPARISON OF THE TESTING ACCURACIES**

Data set	Algorithm	Testing acc. (two-fold CV repeated six times)						Avg.
		1	2	3	4	5	6	acc.
Crude_oil	Our rule-base	85.7	87.5	75.0	83.9	73.2	82.1	<b>81.2</b>
	C5.0	76.8	78.6	80.4	80.4	76.8	75.0	78.0
Glass	Our rule-base	64.0	66.4	65.4	61.2	64.0	63.1	64.0
	C5.0	65.9	67.8	65.0	67.3	66.4	69.6	<b>67.0</b>
Iris	Our rule-base	96.0	93.3	94.7	96.0	95.3	94.0	<b>94.9</b>
	C5.0	92.0	94.7	92.0	92.7	91.3	92.7	92.6
Myo_electric	Our rule-base	81.9	93.1	83.3	91.7	91.7	91.7	<b>88.9</b>
	C5.0	83.3	90.3	79.2	86.1	93.1	88.9	86.8
Norm4	Our rule-base	93.8	94.4	94.4	95.3	94.3	92.5	<b>94.1</b>
	C5.0	89.8	91.3	91.3	90.6	91.8	89.9	90.8
Bupa	Our rule-base	60.9	59.7	64.6	64.1	64.4	64.1	63.0
	C5.0	65.8	62.3	65.8	68.7	63.5	64.0	<b>65.0</b>
Promoters	Our rule-base	76.4	76.4	68.9	76.4	79.3	75.5	<b>75.5</b>
	C5.0	75.5	74.5	69.8	71.7	78.3	78.3	74.7
Heart	Our rule-base	76.7	80.0	77.4	78.2	78.2	77.0	<b>77.9</b>
	C5.0	74.1	77.0	76.3	77.8	79.6	73.3	76.4
Golf	Our rule-base	92.9	67.9	60.7	85.7	82.1	78.6	<b>78.0</b>
	C5.0	82.1	71.4	57.1	71.4	78.6	71.4	72.0
Australian	Our rule-base	84.6	84.1	85.5	84.8	84.4	84.4	<b>84.6</b>
	C5.0	83.2	84.5	85.4	85.8	84.8	83.1	84.5

TABLE X

## COMPARISON OF THE NUMBER OF THE RULES

Data set	Algorithm	# of rules (two-fold CV repeated six times)						Avg. rules
		1	2	3	4	5	6	
Crude_oil	Our rule-base	5.5	5.0	5.5	6.0	5.0	5.5	5.4
	C5.0	4.0	4.0	5.0	4.0	4.5	3.0	<b>4.1</b>
Glass	Our rule-base	20.0	12.5	13.0	15.0	16.0	9.0	14.3
	C5.0	10.0	9.5	13.5	7.0	9.5	9.0	<b>9.8</b>
Iris	Our rule-base	4.5	3.0	4.5	5.0	5.0	5.0	4.5
	C5.0	4.0	3.5	3.0	4.0	3.0	3.0	<b>3.4</b>
Myo_electric	Our rule-base	2.5	2.5	2.5	3.5	2.0	3.5	<b>2.8</b>
	C5.0	3.5	3.0	3.5	4.0	3.5	4.0	3.6
Norm4	Our rule-base	12.0	9.5	17.0	12.0	13.0	10.0	<b>12.3</b>
	C5.0	14.5	14.5	13.5	12.5	11.5	14.5	13.5
Bupa	Our rule-base	5.5	5.0	3.5	7.0	7.0	4.0	<b>5.3</b>
	C5.0	14.0	9.5	17.0	13.0	16.0	11.0	13.4
Promoters	Our rule-base	5.0	1.5	3.5	12.5	8.0	8.5	<b>6.5</b>
	C5.0	9.0	7.0	8.5	8.0	5.5	7.5	7.6
Heart	Our rule-base	16.0	9.5	15.0	14.0	7.0	13.5	12.5
	C5.0	11.0	12.0	12.5	11.5	12.5	11.5	<b>11.8</b>
Golf	Our rule-base	5.0	3.5	4.5	6.0	5.5	6.5	5.2
	C5.0	5.0	4.5	2.5	2.5	3.0	2.5	<b>3.3</b>
Australian	Our rule-base	3.5	3.0	2.5	2.0	3.5	3.5	<b>3.0</b>
	C5.0	8.5	10.5	13.5	11.5	14.0	9.0	11.2

TABLE XI  
COMPARISON OF THE BEST PERFORMANCE

Data set	Our rule-base		C5.0 rule-base	
	# of rules	Testing acc.	# of rules	Testing acc.
Crude_oil	5.0	<b>87.5</b>	4.0	80.4
Glass	12.5	66.4	9.0	<b>69.6</b>
Iris	4.5	<b>96.0</b>	3.5	94.7
Myo_electric	2.5	93.1	3.5	93.1
Norm4	12.0	<b>95.3</b>	11.5	91.8
Bupa	3.5	64.6	13.0	<b>68.7</b>
Promoters	8.0	<b>79.3</b>	5.5	78.3
Heart	9.5	<b>80.0</b>	12.5	79.6
Golf	5.0	<b>92.9</b>	5.0	82.1
Australian	2.5	85.5	11.5	<b>85.8</b>

From the result shown in Table IX, we find that testing accuracy of the two data sets Glass and Bupa by our method is lower than that by C5.0. In order to improve this problem, we focus on these two data sets. We observe these data sets and find that both of them have continuous attributes and discrete attributes. As mention in Chapter 3, we introduce CAIM algorithm to discretize continuous attributes. We use CAIM algorithm to deal with the data sets Glass and Bupa, and then tested by our scheme. Table XII and Table XIII show the performance of our GA based fuzzy ID3 with and without CAIM algorithm.

The performances of these two data sets are illustrated in Table XII and Table XIII. After testing ten times, the average accuracy of our method with CAIM algorithm is superior to that without CAIM algorithm. As demonstrated in the testing, the proposed CAIM algorithm is helpful to improve the testing accuracy.

**TABLE XII**  
**COMPARISON OF THE ACCURACY BY DIFFERENT PROCESS**

	Glass		Bupa	
	ORIGINAL	WITH CAIM	ORIGINAL	WITH CAIM
1	65.4	65.9	73.9	74.2
2	68.7	64.0	72.5	75.1
3	65.9	65.4	74.2	73.6
4	61.7	70.6	75.9	74.5
5	63.0	66.8	73.6	76.8
6	65.9	62.2	74.8	76.2
7	70.0	66.4	72.8	74.2
8	65.9	69.2	76.8	75.0
9	65.4	64.5	68.7	75.6
10	66.4	68.2	73.0	74.8
Avg. acc.	65.8	<b>66.3</b>	73.6	<b>75.0</b>

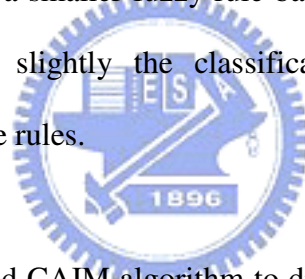
**TABLE XIII**

**COMPARISON OF THE ACCURACY BY DIFFERENT PROCESS (TWO-FOLD CV)**

	Glass		Bupa	
	ORIGINAL	WITH CAIM	ORIGINAL	WITH CAIM
1	61.3	66.4	57.7	66.1
2	63.1	63.6	66.1	64.9
3	60.3	65.4	64.3	67.5
4	64.5	70.1	69.6	59.7
5	66.4	64.2	64.4	66.7
6	63.6	62.2	64.1	65.8
7	64.5	66.4	61.2	64.9
8	65.4	69.2	65.8	65.2
9	66.4	65.4	68.4	66.1
10	65.4	64.0	58.8	64.7
Avg. acc.	64.1	<b>65.7</b>	64.0	<b>65.2</b>

## Chapter 5. Conclusion

In this thesis, we proposed a genetic algorithm based fuzzy ID3 method to construct fuzzy classification system, which can accept continuous, discrete, or mixed-mode data sets. Next, we formulated a rule pruning method to obtain a more efficient rule base. Our proposed method can directly classify mixed-mode data set with high classification accuracy. On testing to some famous data sets, which include continuous, discrete, and mixed-mode data sets, we have obtained very high classification accuracy with small number of rules. It is remarked that the decision tree after pruning can lead to a smaller fuzzy rule base and the pruned rule base can usually remain or decrease slightly the classification performance despite the deduction of the number of the rules.



Furthermore, we proposed CAIM algorithm to discretize the data sets and tested by our GA based fuzzy ID3 method. The performance of the testing accuracy and generated rules by our method with CAIM algorithm is better averagely than that without CAIM algorithm. On comparing the results generated by our proposed method with C5.0, we find that our rule-base outperforms C5.0 in eight out of ten data sets except for the data sets Glass and Bupa. To be directed against the two data sets, we find that the average testing accuracy of our GA based fuzzy ID3 method with CAIM algorithm is superior to that without CAIM algorithm. As demonstrated in the testing, the proposed CAIM algorithm is helpful to improve the testing accuracy.

The features employed in this thesis are independent scalar. We will extend our



GA based fuzzy ID3 method to a vector format. Computation consuming is another task in the field of machine learning, we must try to reduce the computation burden in this scheme. These will be a good challenge to study in the future.



## References

- [1] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets Syst.*, vol. 69, pp. 125–139, 1995.
- [2] M. S. Chen and J. Han, "Data mining: An overview from a database perspective," *IEEE Trans. Knowledge and Data Eng.*, vol. 8, no. 6, pp. 866–883, Dec. 1996.
- [3] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, pp.81–106, 1986.
- [4] J. R. Quinlan, *C4.5, Programs for Machine Learning*. San Mateo, CA: Morgan Kauffman, 1993.
- [5] Data Mining Tools, <http://www.rulequest.com/see5-info.html>, 2003.
- [6] L. Breiman *et al.*, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks/Cole (1984).
- [7] M. Umamo *et al.*, "Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems," in *Proc. Third IEEE Conf. on Fuzzy Systems*, vol. 3, pp. 2113–2118, 1994.
- [8] C. Z. Janikow, "Fuzzy decision trees: issues and methods," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 1, pp. 1–14, Feb. 1998.
- [9] J. Catlett, "On changing continuous attributes into ordered discrete attributes," in *Proc. European Working Session on Learning*, pp. 164-178, 1991.
- [10] J. Y. Ching, A. K. C. Wong, and K. C. C. Chan, "Class-dependent discretization for inductive learning from continuous and mixed mode data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 7, pp. 641–651, July

- 1995.
- [11] L. A. Kurgan and K. J. Cios, "CAIM discretization algorithm," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 2, pp. 145–153, Feb. 2004.
- [12] L. G. Sison and E. K. P. Chong, "Fuzzy modeling by induction and pruning of decision trees," in *Proc. IEEE Int. Symp. Intell. Contr.*, Columbus, OH, Aug. 1994, pp. 166–171.
- [13] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River, New Jersey: Prentice-Hall, 1996.
- [14] N. R. Pal and S. Chakraborty, "Fuzzy rule extraction from ID3-type decision trees for real data," *IEEE Trans. Syst., Man, Cybern B*, vol. 31, no. 5, pp. 745–754, Oct. 2001
- [15] X. Z. Wang and J. R. Hong, "On the handling of fuzziness for continuous-valued attributes in decision tree generation," *Fuzzy Sets Syst.*, vol. 99, pp. 283–290, 1998.
- [16] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, New Jersey, Prentice-Hall, 2002.
- [17] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and Unsupervised Discretization of Continuous Features," in *Proc. 12th Int'l Conf. Machine Learning*, pp. 194–202, 1995.
- [18] C. Blake and E. K. Merz, *UCI Repository of Machine Learning Database*, 1998.
- [19] H. Wang, D. Bell, and F. Murtagh, "Axiomatic approach to feature subset selection based on relevance," *IEEE Trans. Pattern Analysis and Machine Intelligence.*, vol. 21, no. 3, pp. 271–277, March 1999.