ELSEVIER

Fuzzy Sets and Systems 84 (1996) 33-47

FUZZY sets and systems

Induction of fuzzy rules and membership functions from training examples[1]

Tzung-Pei Hong[a,*], Chai-Ying Lee[b]

[a] *Department of Information Management, Kaohsiung Polytechnic Institute, Kaohsiung 84008, Taiwan, ROC*
[b] *Institute of Electrical Engineering, Chung-Hua Polytechnic Institute, Hsinchu, 30067, Taiwan, ROC*

Received January 1995; revised 1 August 1995

Abstract

Most fuzzy controllers and fuzzy expert systems must predefine membership functions and fuzzy inference rules to map numeric data into linguistic variable terms and to make fuzzy reasoning work. In this paper, we propose a general learning method as a framework for automatically deriving membership functions and fuzzy if-then rules from a set of given training examples to rapidly build a prototype fuzzy expert system. Based on the membership functions and the fuzzy rules derived, a corresponding fuzzy inference procedure to process inputs is also developed.

Keywords: Expert systems; Fuzzy clustering; Fuzzy decision rules; Fuzzy machine learning; Knowledge acquisition; Membership functions

1. Introduction

Decision-making is one of the most important activities in the real world. With specific domain knowledge as a background, the task of decision-making is to get an optimal or a nearly optimal solution from input information using an inference procedure. Generally, there are three ways to make a decision in a complex environment:

(1) by building a mathematical model;
(2) by seeking human experts' advice;
(3) by building an expert system or controller.

Among them, building an accurate mathematical model to describe the complex environment is a good way. However, accurate mathematical models neither always exist nor can they be derived for all complex environments because the domain may not be thoroughly understood. The first method is then limited and when it fails, an alternative for making a good decision is to seek human experts' help. However, the cost of querying an expert may be high, and there may be no human experts available when the decision must be made.

Recently, expert systems have been widely used in domains for which the first two methods are not suitable [5, 19]. The knowledge base in an expert system can grow incrementally and can be updated dynamically, so that the performance of an expert

[1] This research was supported by the National Science Council of the Republic of China under contract NSC84-2213-E-214-013.

* Corresponding author. E-mail: tphong@nas05.kpi.edu.tw.

SSDI 0165-0114(95)00305-3

system will become better and better as it develops. Also, the expert system approach can integrate expertise from many fields, reduce the cost of query, lower the probability of danger occurring, and provide fast response [19].

To apply expert systems in decision-making, having the capacity to manage uncertainty and noise is quite important. Several theories such as *fuzzy set theory* [27,29], *probability theory*, *D-S theory* [21], and approaches based on *certainty factors* [2], have been developed to manage uncertainty and noise. Fuzzy set theory is more and more frequently used in expert systems and controllers, because of its simplicity and similarity to human reasoning. The theory has been applied to many fields such as manufacturing, engineering, diagnosis, economics, and others [5,9,28]. However, current fuzzy systems have the following general limitations.

(1) They have no common framework from which to deal with different kinds of problems; in other words, they are problem-dependent.

(2) Human experts play a very important role in developing fuzzy expert systems and fuzzy controllers.

Most fuzzy controllers and fuzzy expert systems can be seen as special rule-based systems that use fuzzy logic. A fuzzy rule-based expert system contains fuzzy rules in its knowledge base and derives conclusions from the user inputs and the fuzzy reasoning process [9,28]. A fuzzy controller is a knowledge-based control scheme in which scaling functions of physical variables are used to cope with uncertainty in process dynamics or the control environment [7]. They must usually predefine membership functions and fuzzy inference rules to map numeric data into linguistic variable terms (e.g. very high, young, ...) and to make fuzzy reasoning work. The linguistic variables are usually defined as fuzzy sets with appropriate membership functions. Recently, many fuzzy systems that automatically derive fuzzy if-then rules from numeric data have been developed [3,13,18,22,23]. In these systems, prototypes of fuzzy rule bases can then be built quickly without the help of human experts, thus avoiding a development bottleneck. Membership functions still need to be predefined, however, and thus are usually built by human experts or experienced users. The same problem as before then arises: if the experts are not available, then the membership functions cannot be accurately defined, or the fuzzy systems developed may not perform well.

In this paper, we propose a general learning method as a framework to derive membership functions automatically and fuzzy if-then rules from a set of given training examples and quickly build a prototype of an expert system. Based on the membership functions and the fuzzy rules derived, a corresponding fuzzy inference procedure to process the input in question is developed.

The remaining parts of this paper are organized as follows. In Section 2, the development of an expert system is introduced. In Section 3, the concept and terminology of fuzzy sets are briefly reviewed. In Section 4, the basic architecture of fuzzy expert systems is provided. In Section 5, a new learning method is given for automatically deriving membership functions and fuzzy if-then rules from a set of training instances. In Section 6, an inference procedure for processing inputs based on the derived rules is suggested. In Section 7, application to Fisher's iris data is presented, Conclusions are given in Section 8.

2. Development of an expert system

Development of a classical expert system is illustrated in Fig. 1 [19]. A knowledge engineer first establishes a dialog with a human expert in order to elicit the expert's knowledge. The knowledge engineer then encodes the knowledge for entry into the knowledge base. The expert then evaluates the expert system and gives a critique to the knowledge engineer. This process continues until the system's performance is judged to be satisfactory by the expert. The user then supplies facts or other information to the expert system and receives expert advice in response [19].

Although a wide variety of expert systems have been built, a development bottleneck occurs in knowledge acquisition. Building a large-scale expert system involves creating and extending a large

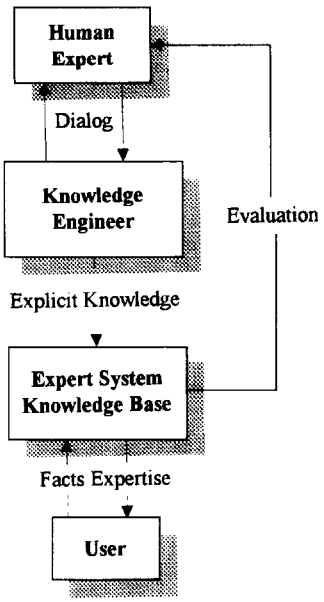

Fig. 1. Development of a classical expert system.

knowledge base over the course of many months or years. For instance, the knowledge base of the XCON (RI) expert system has grown over the past 10 years from 300 component descriptions and 750 rules to 31 000 component descriptions and 10 000 rules [20]. Shortening the development time is then the most important factor for the success of an expert system. Recently, machine-learning techniques have been developed to ease the knowledge-acquisition bottleneck. Among machine-learning approaches, deriving inference rules from training examples is the most common [12, 16, 17]. Given a set of examples and counterexamples of a concept, the learning program tries to induce general rules that describe all of the positive training instances and none of the counter-examples. If the training instances belong to more than two classes, the learning program tries to induce general rules that describe each class. In addition to classical machine learning methods, fuzzy learning methods (such as fuzzy ID3) [8, 24, 25] for inducing fuzzy knowledge have also emerged recently. Machine learning then provides a feasible way to build a prototype (fuzzy) expert system.

3. Review of fuzzy set theory

Fuzzy set theory was first proposed by Zadeh in 1965 [26], and was first used in control by Mamdani [15]. Fuzzy set theory is primarily concerned with quantifying and reasoning using natural language in which many words have ambiguous meanings. It can also be thought of as an extension of the traditional crisp set, in which each element is either in the set or not in the set.

Formally, the process by which individuals from a universal set X are determined to be either members or non-members of a crisp set can be defined by a *characteristic* or *discrimination function* [11]. For a given crisp set A, this function assigns a value $\mu_A(x)$ to every $x \in X$ such that

$$\mu_A(x) = \begin{cases} 1 & \text{if and only if } x \in A, \\ 0 & \text{if and only if } x \notin A. \end{cases} \tag{1}$$

Thus, the function maps elements of the universal set to the set containing 0 and 1. This can be indicated by

$$\mu_A : X \to \{0, 1\}. \tag{2}$$

This kind of function can be generalized such that the values assigned to the elements of the universal set fall within a specified range and are referred to as the membership grades of these elements in the set. Larger values denote higher degrees of set membership. Such a function is called a membership function μ_A by which a fuzzy set A is usually defined. This function can be indicated by

$$\mu_A : X \to [0, 1], \tag{3}$$

where X refers to the universal set defined in a specific problem, and $[0, 1]$ denotes the interval of real numbers from 0 to 1, inclusively.

Assume that A and B are two fuzzy sets with membership functions of μ_A and μ_B. The most commonly used primitives for fuzzy union and fuzzy intersection are as follows [23]:

$$\begin{aligned} \mu_{A \cup B} &= \max\{\mu_A, \mu_B\}, \\ \mu_{A \cap B} &= \min\{\mu_A, \mu_B\}. \end{aligned} \tag{4}$$

Although these two operations may cause the problems of *partially single operand dependency* and *negative compensation* [10], they are the most

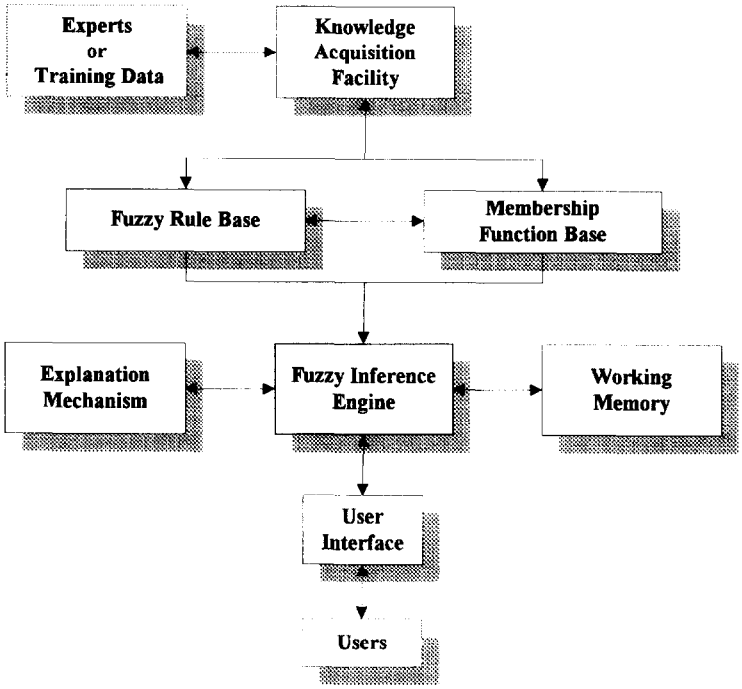

Fig. 2. Architecture of a fuzzy expert system.

commonly used because of their simplicity. These two operators are also used in this paper in deriving the fuzzy if-then rules and membership functions.

4. Architecture of a fuzzy expert system

Fig. 2 shows the basic architecture of a fuzzy expert system. Individual components are illustrated as follows.

- *User interface*: For communication between users and the fuzzy expert system. The interface should be as friendly as possible.
- *Membership function base*: A mechanism that presents the membership functions of different linguistic terms.
- *Fuzzy rule base*: A mechanism for storing fuzzy rules as expert knowledge.
- *Fuzzy inference engine*: A program that executes the inference cycle of fuzzy matching, fuzzy conflict resolution, and fuzzy rule-firing according to given facts.
- *Explanation mechanism*: A mechanism that explains the inference process to users.
- *Working memory*: A storage facility that saves user inputs and temporary results.
- *Knowledge-acquisition facility*: An effective knowledge-acquisition tool for conventional interviewing or automatically acquiring the expert's knowledge, or an effective machine-learning approach to deriving rules and membership functions automatically from training instances, or both.

Here the membership functions are stored in a knowledge base (instead of being put in the interface) since by our method, decision rules and membership functions are acquired by a learning method. When users input facts through the user interface, the fuzzy inference engine automatically reasons using the fuzzy rules and the membership functions, and sends fuzzy or crisp results through the user interface to the users as outputs.

In the next section, we propose a general learning method as a knowledge-acquisition facility for automatically deriving membership functions and

fuzzy rules from a given set of training instances. Based on the membership functions and the fuzzy rules derived, a corresponding fuzzy inference procedure to process user inputs is developed.

5. The knowledge acquisition facility

A new learning method for automatically deriving fuzzy rules and membership functions from a given set of training instances is proposed here as the knowledge acquisition facility. Notation and definitions are introduced below.

5.1. Notation and definitions

In a training instance, both input and desired output are known. For a m-dimensional input space, the ith training example can then be described as

$$(x_{i1}, x_{i2}, \ldots, x_{im}; y_i), \tag{5}$$

where $x_{ir} (1 \leqslant r \leqslant m)$ is the rth attribute value of the ith training example and y_i is the output value of the ith training example.

For example, assume an insurance company decides *insurance fees* according to two attributes: *age* and *property*. If the insurance company evaluates and decides the insurance fee for a person of age 20 possessing property worth \$30 000 should be \$1000, then the example is represented as (age = 20, property = \$30 000, insurance fee = \$1000).

5.2. The algorithm

The learning activity is shown in Fig. 3 [23].

A set of training instances are collected from the environment. Our task here is to generate automatically reasonable membership functions and appropriate decision rules from these training data, so that they can represent important features of the data set. The proposed learning algorithm can be divided into five main steps.

Step 1: cluster and fuzzify the output data;

Step 2: construct initial membership functions for input attributes;

Step 3: construct the initial decision table;

Step 4: simplify the initial decision table;

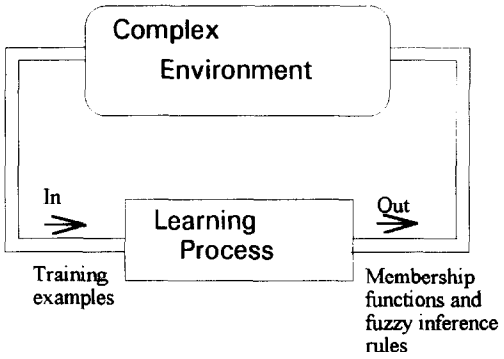

Fig. 3. Learning activity.

Step 5: rebuild membership functions in the simplification process;

Step 6: derive decision rules from the decision table.

Details are illustrated by an example in the next section.

5.3. Example

As before, assume an insurance company decides *insurance fees* according to *age* and *property*. Each training instance then consists of two attributes: *age* and *property* (in ten thousands), and one output: *insurance fee*. The goal of the learning process is to construct a membership function for each attribute (i.e. age and property), and to derive fuzzy decision rules to decide on reasonable insurance fees.

Assume the following eight training examples are available:

Age	*Property*	*Insurance fee*
(20,	30;	2000)
(25,	30;	2100)
(30,	10;	2200)
(45,	50;	2500)
(50,	30;	2600)
(60,	10;	2700)
(80,	30;	3200)
(80,	40;	3300)

The learning algorithm proceeds as follows.

Step 1: *Cluster and fuzzify the output data.* In this step, the output values of all training instances

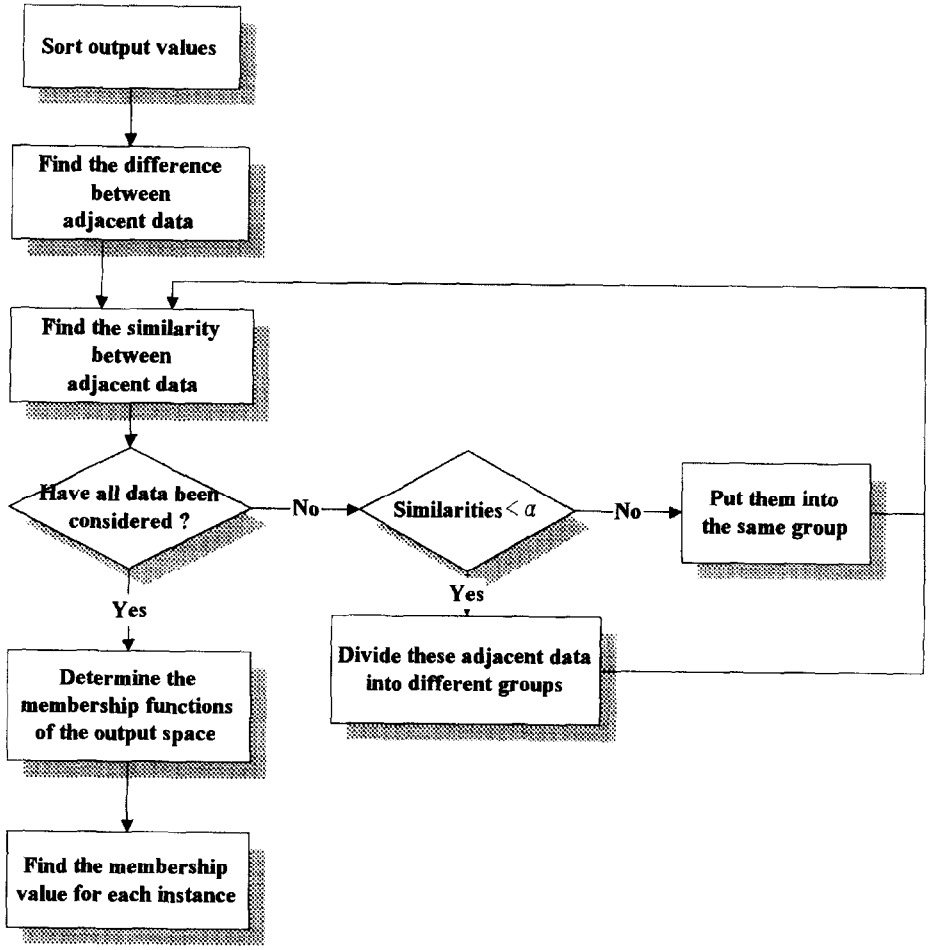

Fig. 4. The flow chart of Step 1.

are appropriately grouped by applying the clustering procedure below, and appropriate membership functions for output values are derived. Our clustering procedure considers training instances with close output values as belonging to the same class with high membership values. Six substeps are included in Step 1. The flow chart is shown in Fig. 4. Details are as follows.

Substep (1a): *Sort the output values of the training instances in an ascending order.* It sorts the output values of the training instances to find the relationship between adjacent data. The modified order after sorting is then

$$y'_1, y'_2, \ldots, y'_n. \tag{6}$$

where $y'_i \leqslant y'_{i+1}$ (for $i = 1, \ldots, n-1$).

Example 1. For the training instances given in the example, Substep (1a) proceeds as follows.

Original order:

2000, 2100, 2200, 2500, 2600, 2700, 3200, 3300

↓ sorting

Modified order:

2000, 2100, 2200, 2500, 2600, 2700, 3200, 3300

Substep (1b): *Find the difference between adjacent data.* The difference between adjacent data provides the information about the similarity between them. For each pair y'_i and y'_{i+1} ($i = 1, 2, \ldots, n-1$), the difference is $diff_i = y'_{i+1} - y'_i$.

Example 2. From the result of Substep(1a), the difference sequence is calculated as follows.

Modified order:
2000, 2100, 2200, 2500, 2600, 2700, 3200, 3300,

Difference sequence:
100, 100, 300, 100, 100, 500, 100.

Substep (1c): *Find the value of similarity between adjacent data.* In order to obtain the value of similarity between adjacent data, we convert each distance $diff_i$ to a real number s_i between 0 and 1 according to the following formula [6]:

$$s_i = \begin{cases} 1 - \dfrac{diff_i}{C * \sigma_s} & \text{for } diff_i \leqslant C * \sigma_s, \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where s_i represents the similarity between y'_i and y'_{i+1}, $diff_i$ is the distance between y'_i and y'_{i+1}, σ_s is the standard derivation of $diff_i$'s, and C is a control parameter deciding the shape of the membership functions of similarity. A larger C causes a greater similarity.

Example 3. Assume the control parameter $C = 4$. The standard deviation σ is first calculated as 145.69. Each membership value of similarity is shown as follows:

$s_1 = 1 - 100/145.69 * 4 = 0.83,$

$s_2 = 1 - 100/145.69 * 4 = 0.83,$

$s_3 = 1 - 300/145.69 * 4 = 0.49,$

$s_4 = 1 - 100/145.69 * 4 = 0.83,$

$s_5 = 1 - 100/145.69 * 4 = 0.83,$

$s_6 = 1 - 500/145.69 * 4 = 0.14,$

$s_7 = 1 - 100/145.69 * 4 = 0.83.$

Substep (1d): *Cluster the training instances according to similarity.* Here we use the α-cut of similarity to cluster the instances. The value of α determines the threshold for two adjacent data to be thought of as belonging to the same class. Larger α will have a smaller number of groups. The procedure proceeds as follows:

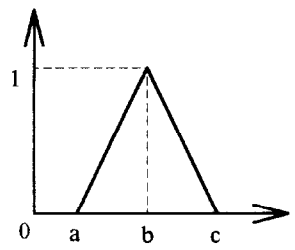

Fig. 5. A triangle membership function.

If $s_i < \alpha$ **then** divide the two adjacent data into different groups;
else put them into the same group.

After the above operation, we can obtain the result formed as (y'_i, R_j), meaning that the ith output data will be clustered into the R_j, where R_j means the jth produced fuzzy region.

Example 4. Assume α is set at 0.8. The training examples are then grouped as follows:

$$(y'_1, R_1), (y'_2, R_1), (y'_3, R_1) / (y'_4, R_2), (y'_5, R_2), (y'_6, R_2)$$

$$/ (y'_7, R_3), (y'_8, R_3).$$

Substep (1e): *Determine membership functions of the output space.* For simplicity, triangle membership functions are used here for each linguistic variable. A triangle membership function can be defined by a triad (a, b, c) as Fig. 5 shows (not necessarily symmetric).

In this substep, we propose a heuristic method for determining these three parameters. First, we assume that the center point b lies at the center-of-gravity of the group. Next, we try to find the membership values of two boundary training outputs in the group, where "boundary training outputs" mean the minimum and maximum outputs in that group. The two end points a and c of the output membership function can then be found through the extrapolation of b and the two boundary training outputs. The following four procedures are then used to achieve this purpose.

Procedure 1: Find the central-vertex-point b_j:
if $y'_i, y'_{i+1}, \ldots, y'_k$ belong to the jth group,
then the central-vertex-point b_j in this group is defined as

$$b_j = \frac{y'_i * s_i + y'_{i+1} * \frac{s_i + s_{i+1}}{2} + y'_{i+2} * \frac{s_{i+1} + s_{i+2}}{2} + \cdots + y'_{k-1} * \frac{s_{k-2} + s_{k-1}}{2} + y'_k * s_{k-1}}{s_i + \frac{s_i + s_{i+1}}{2} + \frac{s_{i+1} + s_{i+2}}{2} + \cdots + \frac{s_{k-2} + s_{k-1}}{2} + s_{k-1}}. \quad (8)$$

Procedure 2: Determine the membership of y'_i and y'_k.

The minimum similarity in the group is chosen as the membership value of the two boundary points y'_i and y'_k. Restated, the following formulas are used to calculate $\mu_j(y'_i)$ and $\mu_j(y'_k)$, where μ_j represents the membership of belonging to the jth group:

$$\mu_j(y'_i) = \mu_j(y'_k) = \min(s_i, s_{i+1}, \ldots, s_{k-1}). \quad (9)$$

This heuristic is quite rational since the two boundary instances should be clustered with the group having a lower probability than other instances in the same group.

Procedure 3: Determine the vertex-point a:
According to the two points $(b_j, 1)$ and $(y'_i, \mu_j(y'_i))$, we can find the point $(a, 0)$ by interpolation as follows:

$$a = b_j - \frac{b_j - y'_i}{1 - \mu_j(y'_i)}. \quad (10)$$

Procedure 4: Determine the vertex-point c.
According to the two points $(b_j, 1)$ and $(y'_k, \mu_j(y'_k))$, we can find the point $(c, 0)$ by interpolation as follows:

$$c = b_j + \frac{y'_k - b_j}{1 - \mu_j(y'_k)}. \quad (11)$$

Example 5. After the operation of Substep(1e), the membership functions of the three output groups are derived as shown in Fig. 6.

Note that the membership functions may not necessarily overlap each other. The shape of the membership functions is affected by the choice of C. As mentioned above, a larger C will cause a greater similarity. The two boundary instances in a group then have larger similarities, causing the membership function formed to be flatter.

Substep (1f): *Find the membership value of belonging to the desired group for each instance.*
From the above membership functions we can get the fuzzy value of each output data formed as (y'_i, R_j, μ_{ij}), referred to as the ith output data has the fuzzy value μ_{ij} to the cluster R_j. Each training instance is then transformed as

$$(x_1, x_2, \ldots, x_m; (R_1, \mu_1), (R_2, \mu_2), \ldots, (R_k, \mu_k)). \quad (12)$$

Example 6. Each output is transformed as follows:

$(y'_1, R_1, 0.83), (y'_2, R_1, 1), (y'_3, R_1, 0.83), (y'_4, R_2, 0.83),$

$(y'_5, R_2, 1), (y'_6, R_2, 0.83), (y'_7, R_3, 0.83), (y'_8, R_3, 0.83).$

Combining with the input data, we obtain
(20, 30; $(R_1, 0.83)$),
(25, 30; $(R_1, 1)$),
(30, 10; $(R_1, 0.83)$),
(45, 50; $(R_2, 0.83)$),
(50, 30; $(R_2, 1)$),
(60, 10; $(R_2, 0.83)$),
(80, 30; $(R_3, 0.83)$),
(80, 40; $(R_3, 0.83)$).

Step 2: Construct initial membership functions for input attributes. We assign each input attribute an initial membership function, which is assumed

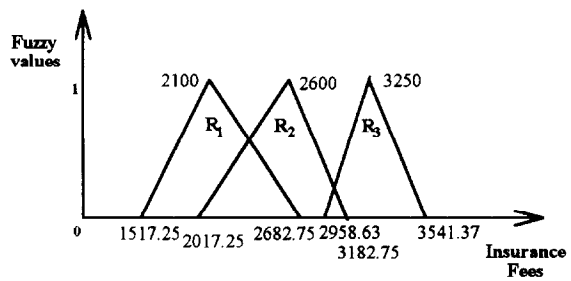

Fig. 6. Insurance fee membership functions.

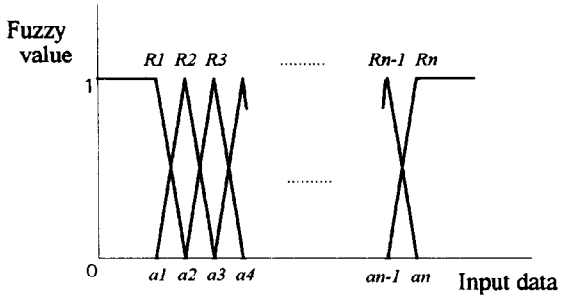

Fig. 7. Initial membership functions.

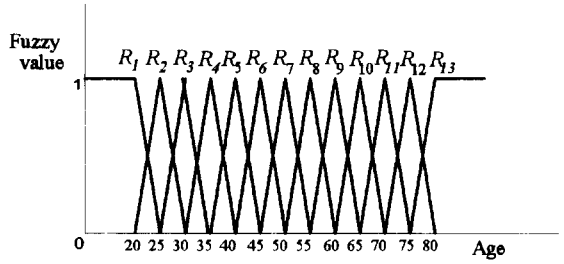

Fig. 8. Initial membership functions of age.

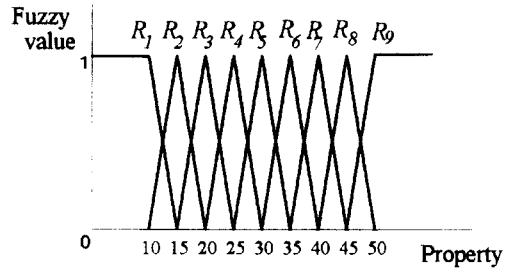

Fig. 9. Initial membership functions of property.

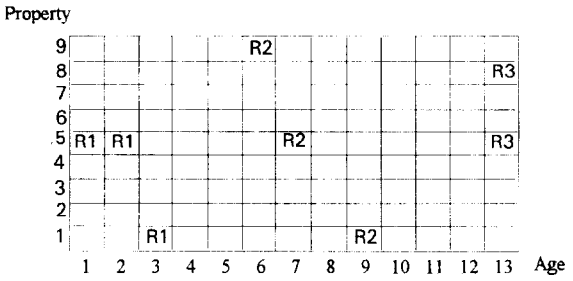

Fig. 10. Initial decision table for insurance problem.

to be a triangle (a, b, c) with $b - a = c - b =$ the smallest predefined unit. For example, if three values of an attribute are 10, 15 and 20, then the smallest unit is chosen to be 5. Here we let a_0 be the smallest value for the attribute and a_n be the biggest value for the attribute. Initial membership functions for the attribute are viewed as in Fig. 7, where $a_i - a_{i-1} = a_{i+1} - a_i =$ the smallest predefined unit, and R_x means the xth initial region $(i = 2, 3, \ldots, n-1; x = 1, 2, \ldots, n)$.

At first sight, this definition seems unsuitable for problems with small units over big ranges (e.g. 1, 1.1, 1000), since many initial membership functions may exist. But in practical implementation, only the membership functions corresponding to existing attribute values are kept and considered (through an appropriate data structure [14]). The membership functions corresponding to no attribute values are not kept here since they will be merged in later steps.

Example 7. Let 5 be the smallest predefined unit of age and property. The initial membership functions for age are shown in Fig. 8, and for property in Fig. 9.

Step 3: *Construct the initial decision table.* In this step we build a multi-dimensional decision table (each dimension represents a corresponding attribute) according to the initial membership functions. Let a *cell* be defined as the contents of a position in the decision table. $Cell_{(d_1, d_2, \ldots, d_m)}$ then represents the contents of the position $(d_1, d_2, \ldots, d_i, \ldots, d_m)$ in the decision table, where m is the dimension of the decision table and d_i is the position value at the ith dimension. Each cell in the table may be empty, or may contain a fuzzy region (with maximum membership value) of the output data. Again, in practical implementation, only the non-null cells are kept (through an appropriate data structure [14]).

Example 8. The initial decision table for the insurance fee problem is shown in Fig. 10.

Step 4: *Simplify the initial decision table.* In this step, we simplify the initial decision table to eliminate redundant and unnecessary cells. The five merging operations defined here achieve this purpose.

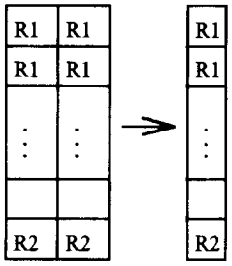

Fig. 11. An example of Operation 1.

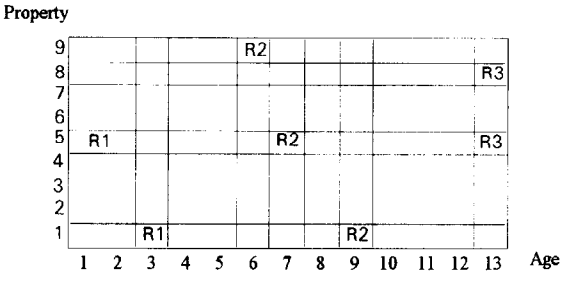

Fig. 12. The results after Operation 1.

Operation 1: If cells in two adjacent columns (or rows) are the same, then merge these two columns (or rows) into one (see Fig. 11). For example, in Fig. 10, all the *cells* in the neighboring columns *age* = 1 and *age* = 2 are the same. The two columns *age* = 1 and *age* = 2 are then merged into one.

Example 9. The decision table after merge operation 1 is shown in Fig. 12.

Operation 2: If two cells are the same or if either of them is empty in two adjacent columns (or rows) and at least one cell in both the columns (or rows) is not empty, then merge these two columns (or rows) into one (see Fig. 13). For example, in Fig. 12, the two columns *age* = 6 and *age* = 7 are merged into one.

Example 10. The decision table after merge operation 2 is shown in Fig. 14.

Operation 3: If all cells in a column (or row) are empty and if cells in its two adjacent columns (or rows) are the same, then merge these three columns (or rows) into one (see Fig. 15).

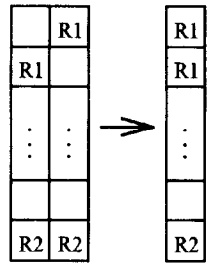

Fig. 13. An example of Operation 2.

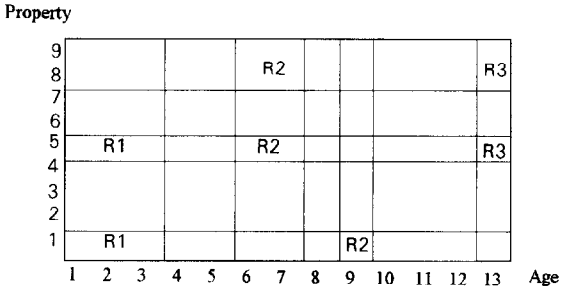

Fig. 14. The results after Operation 2.

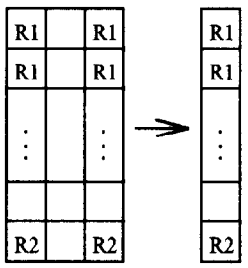

Fig. 15. An example of Operation 3.

Operation 4: If all cells in a column (or row) are empty and if cells in its two adjacent columns (or rows) are the same or either of them is empty, then merge these three columns or rows into one (see Fig. 16). For example, in Fig. 14, the three columns *age* = 7, *age* = 8 and *age* = 9 are merged into one.

Example 11. The decision table after merge operations 3 and 4 is shown in Fig. 17.

Operation 5: If all cells in a column (or row) are empty and if all the non-empty cells in the column (o row) to its left have the same region, and all the

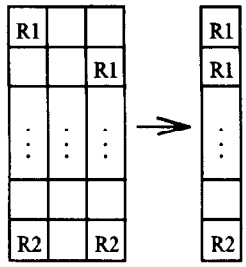

Fig. 16. An example of Operation 4.

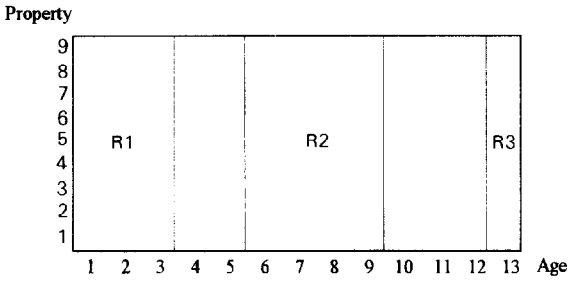

Fig. 17. The results after Operations 3 and 4.

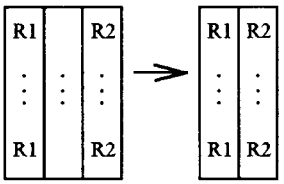

Fig. 18. An example of Operation 5.

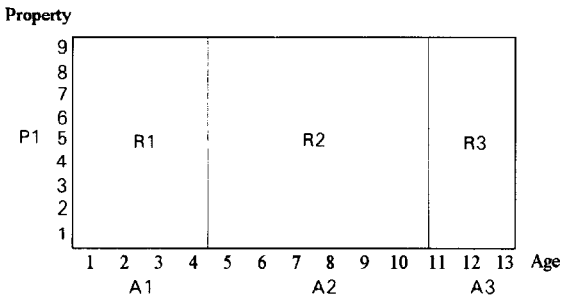

Fig. 19. The results after Operation 5.

non-empty cells in the column (or row) to its right have the same region, but one different from the previously mentioned region, then merge these three columns into two parts (see Fig. 18). For

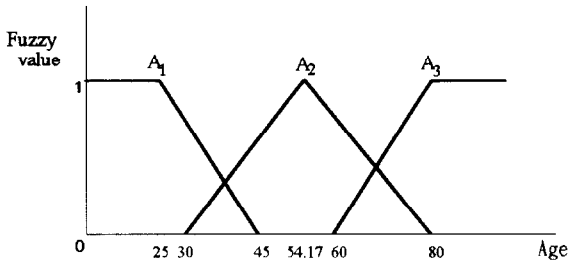

Fig. 20. Final membership functions for age.

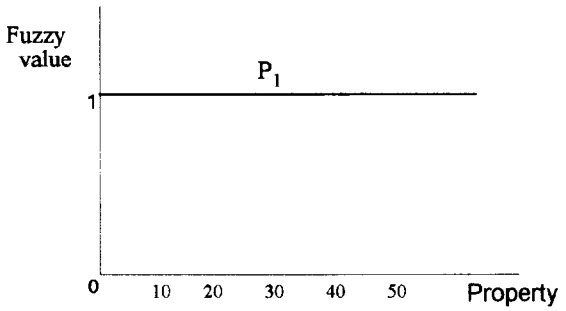

Fig. 21. Final membership function for property.

example, in Fig. 17, the three columns $age = 1$, $age = 4$ and $age = 6$ are then merged into two columns.

Example 12. The decision table after merge operation 5 is shown in Fig. 19.

Step 5: *Rebuild membership functions.* When each merging operation is executed, a corresponding rebuilding of the membership functions for the dimension to be processed is accomplished. For Operations 1 and 2, if (a_i, b_i, c_i) and (a_j, b_j, c_j) are the membership functions for the ith attribute being d_i and d_{i+1}, then the new membership function is $(a_i, (b_i + b_j)/2, c_j)$. For Operations 3 and 4, if (a_i, b_i, c_i), (a_j, b_j, c_j), and (a_k, b_k, c_k) are the membership functions for the ith attribute being $d_i - 1, d_i$ and $d_i + 1$, then the new membership function is $(a_i, (b_i + b_j + b_k)/3, c_k)$. For Operation 5, if (a_i, b_i, c_i), (a_j, b_j, c_j), and (a_k, b_k, c_k) are the membership functions for the ith attribute being $d_i - 1, d_i$ and $d_i + 1$, then the new membership functions are (a_i, b_i, c_j) and (a_j, b_k, c_k).

Example 13. The membership functions as rebuilt after Step 5 are shown in Figs. 20 and 21, respectively.

Step 6: *Derive decision rules from the decision table.* In this step, fuzzy if-then rules are derived from the decision table. Each cell $cell_{(d_1, d_2, \ldots, d_m)} = R_i$ in the decision table is used to derive a rule:

If $input_1 = d_1, input_2 = d_2, \ldots, and\ input_m = d_m$

Then $output = R_i$. (13)

Example 14. After Step 6 we obtain the following rules:

If age is A_1 and property is P_1
Then insurance fee is $R_1 \Rightarrow$ **Rule**$_1$,

If age is A_2 and property is P_1
Then insurance fee is $R_2 \Rightarrow$ **Rule**$_2$,

If age is A_3 and property is P_1
Then insurance fee is $R_3 \Rightarrow$ **Rule**$_3$.

From the example, it is easily seen that the attribute *property* is unimportant in determining the insurance fee. The proposed method then also possesses the capability of conventional machine learning approaches.

6. The fuzzy inference process

In the fuzzy inference process, data are collected from the complex environment, and are used to derive decisions through the fuzzy inference rules and the membership functions.

One advantage of the proposed learning method is that the format of the knowledge derived from the learning process is consistent with that of commonly used fuzzy knowledge, thus the ordinary Mamdani type of inference process [15] can then be applied. The inference process required to obtain a conclusion from the input space is as follows:

Step 1: Convert numeric input values to linguistic terms according to the membership functions derived;

Step 2: Match the linguistic terms with the decision rules to find the output groups;

Step 3: Defuzzify the output groups to form the final decision.

Details are explained below.

Step 1: *Convert numeric input values to linguistic terms according to the membership functions derived.* We map a numeric input $(I_1, I_2, \ldots, I_m)$ to their corresponding fuzzy linguistic terms with membership values.

Example 15. Assume a new input datum $I(37, 40)$ (i.e. age = 37, property = 40) is fed into the fuzzy inference process. It is first converted into the following fuzzy terms through the membership functions derived:

$\mu_{A_1}(I) = 0.55,$

$\mu_{A_2}(I) = 0.3,$

$\mu_{P_1}(I) = 1.$

Step 2: *Match the linguistic terms with the decision rules to find the output groups.* In this step, we match the fuzzy inputs with the fuzzy rules in order to obtain the output regions.

Example 16. The fuzzy input in Example 15 matches the following two rules:

If age is A_1 and property is P_1
Then insurance fee is $R_1 \Rightarrow$ **Rule**$_1$,

If age is A_2 and property is P_1
Then insurance fee is $R_2 \Rightarrow$ **Rule**$_2$.

The membership values of match to Rule 1 and Rule 2 are, respectively, calculated as follows:

$\mu_{R_1}(I) = \min(\mu_{A_1}(I), \mu_{P_1}(I)) = 0.55,$

$\mu_{R_2}(I) = \min(\mu_{A_2}(I), \mu_{P_1}(I)) = 0.3.$

Step 3: *Defuzzify the output groups to form the final output values.* The final output value is obtained by averaging the output groups [29]. Let the membership function for the output group R_i be (a_i, b_i, c_i). The output value is calculated by the following formula:

$$y = \frac{\sum_{i=1}^{K} \mu_{R_i}(I) * b_i}{\sum_{i=1}^{K} \mu_{R_i}(I)}, \tag{14}$$

where K is the number of possible output groups.

Table 1
Testing training examples

Training examples	Testing result	Error rate %	Average error rate (%)
(20, 30; 2000)	2100	5	
(25, 30; 2100)	2100	0	
(30, 10; 2200)	2185.704	0.65	
(45, 50; 2500)	2442.818	2.28	
(50, 30; 2600)	2528.522	2.75	1.94
(60, 10; 2700)	2746.709	1.73	
(80, 30; 3200)	3250	1.56	
(80, 40; 3300)	3250	1.52	

Example 17. The output value for Example 15 is calculated as follows:

$$y = \frac{0.55 * 2100 + 0.3 * 2600}{0.55 + 0.3} = 2276.47.$$

If we have the original training examples as the testing data, then we can derive the results shown in Table 1. From Table 1 we can see that the testing results are quite close to the outputs of the training data. The average error rate is only 1.94%. This is further proof that our membership function is rational.

7. Experimental results

To demonstrate the effectiveness of the proposed fuzzy learning algorithm, we used it to classify Fisher's Iris Data containing 150 training instances [4]. There are three species of iris flowers to be distinguished: setosa, versicolor, and verginica. There are 50 training instances for each class. Each training instance is described by four attributes: sepal width (SW), sepal length (SL), petal width (PW), and petal length (PL). The unit for all four of the attributes is centimeters, measured to the nearest millimeter.

The data set was first split into a training set and a test set, and the fuzzy learning algorithm

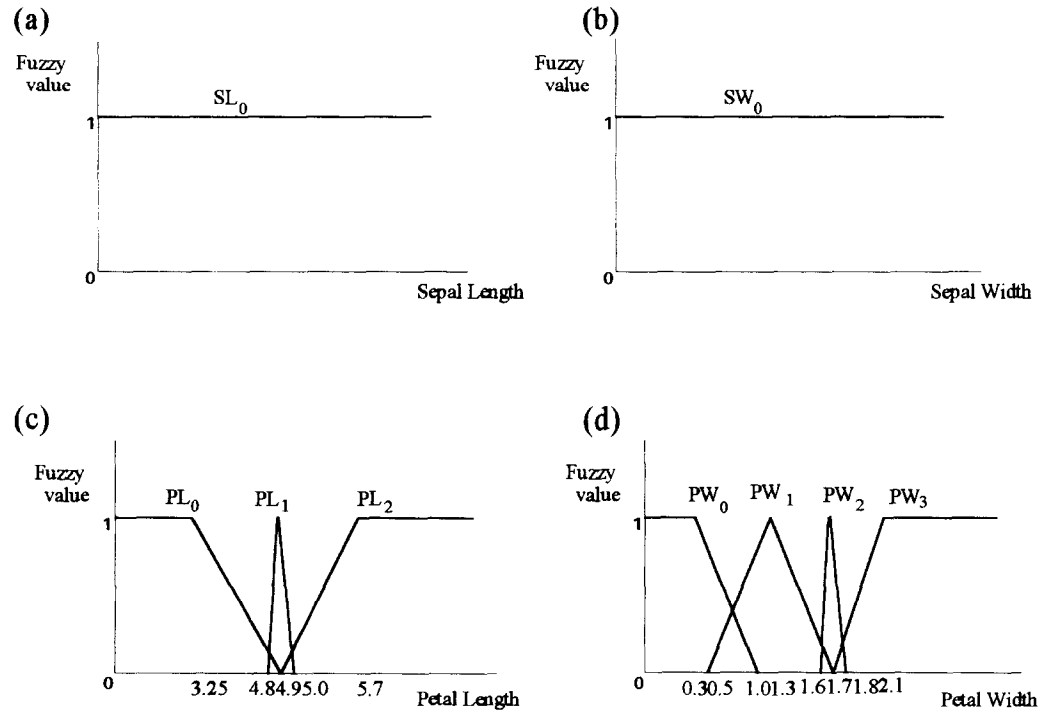

Fig. 22. The derived membership functions of the four attributes.

Table 2
The eight derived fuzzy inference rules

Sepal length	Sepal width	Petal length	Petal width	Class
SL0	SW0	PL0	PW0	Setosa
SL0	SW0	PL0	PW1	Versicolor
SL0	SW0	PL1	PW1	Versicolor
SL0	SW0	PL0	PW3	Versicolor
SL0	SW0	PL2	PW1	Virginica
SL0	SW0	PL0	PW2	Virginica
SL0	SW0	PL1	PW3	Virginica
SL0	SW0	PL2	PW3	Virginica

Table 3
The average accuracy of the fuzzy learning algorithm for the Iris Problem

Setosa	Versicolor	Virginica	Average	Number of rules
100%	94%	92.72%	95.57%	6.21

was run on the training set to induce fuzzy classification rules and membership functions. The rules and membership functions derived were then tested on the test set to measure the percentage of correct predictions. In each run, 50% of the Iris Data were selected at random for training, and the remaining 50% of the data were used for testing.

In the original data order, the derived membership functions of the four attributes are shown in Fig. 22 and the eight derived fuzzy inference rules are shown in Table 2.

From Fig. 22, it is easily seen that the numbers of membership functions for the attributes sepal length and sepal width are one, showing that these two attributes are useless in classifying the Iris Data. Also, the initial membership functions of the attribute petal length were finally merged into only three ranges, and the initial membership functions of the attribute petal width were finally merged into only four ranges, showing that a small number of membership functions are enough for a good reasoning result.

Experiments were then made to verify the accuracy of the fuzzy learning algorithm. For each kind of flower, the correct classification ratio was measured by averaging 200 runs. The number of derived rules was also recorded. Experimental results are shown in Table 3.

From Table 3, it can be seen that high accuracy was obtained from the fuzzy learning algorithm: 100% accuracy for Setosa, 94% for Versicolor and 92.72% for Virginica. Also, the average number of rules in this experiments is 6.21, a small number compared to the number of instances. The fuzzy learning algorithm can then be said to work well in automatically deriving membership functions and inference rules.

8. Conclusions

In this paper, we have proposed a general learning method for automatically deriving membership functions and fuzzy if-then rules from a set of given training examples. The proposed approach can significantly reduce the time and effort needed to develop a fuzzy expert system. Based on the membership functions and the fuzzy rules derived, a corresponding fuzzy inference procedure to process inputs was also applied. Using the Iris Data, we found our model gives a rational result, few rules, and high performance.

Acknowledgements

The authors would like to thank the anonymous referees for their very constructive comments.

References

[1] H.R. Berenji, Fuzzy logic controller, in: R.R. Yager and L.A. Zadeh, Eds., *An Introduction to Fuzzy Logic Applications in Intelligent Systems* (Kluwer Academic Publishers, Dordrecht, 1992) 45–96.

[2] B.G. Buchanan and E.H. Shortliffe, *Rule-Based Expert System: The MYCIN Experiments of the Standford Heuristic Programming Projects* (Addison-Wesley, Reading, MA, 1984).

[3] D.G. Burkhardt and P.P. Bonissone, Automated fuzzy knowledge base generation and tuning, *IEEE Internat. Conf. on Fuzzy Systems* (San Diego, 1992) 179–188.

[4] R. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugenics* 7 (1936) 179–188.

[5] I. Graham and P.L. Jones, *Expert Systems – Knowledge, Uncertainty and Decision* (Chapman and Computing, Boston, 1988) 117–158.

[6] K. Hattori and Y. Tor, Effective algorithms for the nearest neighbor method in the clustering problem, *Pattern Recognition* **26** (1993) 741–746.

[7] S. Isaka and A.V. Sebald, An optimization approach for fuzzy controllers design, *IEEE Trans. Systems Man Cybernet.* **22** (1992) 1469–1473.

[8] O. Itoh, H. Migita and A. Miyamoto, A method of design and adjustment of fuzzy control rules based on operation know-how, *Proc. 3rd IEEE Conf. on Fuzzy Systems* (Orlando, FL, 1994) 492–497.

[9] A. Kandel, *Fuzzy Expert Systems* (CRC Press, Boca Raton, FL, 1992) 8–19.

[10] M.H. Kim, J.H. Lee and Y.J. Lee, Analysis of fuzzy operators for high quality information retrieval, *Inform. Processing Lett.* **46** (1993) 251–256.

[11] G.J. Klir and T.A. Folger, *Fuzzy Sets, Uncertainty, and Information* (Prentice Hall, Englewood Cliffs, NJ, 1992) 4–14.

[12] Y. Kodratoff and R.S Michalski, *Machine Learning: An Artificial Intelligence Approach*, Vol. 3 (Morgan Kaufmann, San Mateo, CA, 1990).

[13] C.C. Lee, Fuzzy logic in control system: fuzzy logic controller – Part I and Part II, *IEEE Trans. Systems Man Cybernet.* **20** (1990) 404–435.

[14] C.Y. Lee, Automatic acquisition of fuzzy knowledge, Master Thesis, (Chung-Hua Polytechnic Institute, Hsinchu, Taiwan, 1995).

[15] E.H. Mamdani, Applications of fuzzy algorithms for control of simple dynamic plant, *IEEE Proc.* **121** (1974) 1585–1588.

[16] R.S. Michalski, J.G. Carbonell and T.M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. 1 (Morgan Kaufmann, Los Altos, CA, 1983).

[17] R.S. Michalski, J.G. Carbonell and T.M. Mitchell, *Machine Learning: An Artificial Intelligence Approach*, Vol. 2 (Morgan Kaufmann, Los Altos, CA, 1984).

[18] H. Nomura, I. Hayashi and N. Wakami, A learning method of fuzzy inference rules by descent method, *IEEE Internat. Conf. on Fuzzy Systems* (San Diego, 1992) 203–210.

[19] G. Riley, *Expert Systems – Principles and Programming* (Pws-Kent, Boston, 1989) 1–59.

[20] J.C. Schimmer, Database consistency via inductive learning, *Proc. 8th Internat. Workshop on Machine Learning* (Morgan Kaufmann, San Mateo, CA, 1991).

[21] G. Shafer and R. Logan, Implementing Dempster's rule for hierarchical evidence, *Artificial Intelligence* **33** (1987) 271–298.

[22] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems Man Cybernet.* **15** (1985) 116–132.

[23] L.X. Wang and J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Trans. Systems Man Cybernet.* **22** (1992) 1414–1427.

[24] R. Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, *Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks* (Iizuka, Japan, 1992) 265–268.

[25] Y. Yuan and M.J. Shaw, Induction of fuzzy decision trees, *Fuzzy Sets Systems* **69** (1995) 125–139.

[26] L.A. Zadeh, *Fuzzy Sets, Inform. and Control* **8** (1965) 338–353.

[27] L.A. Zadeh, Fuzzy logic, *IEEE Comput.* (1988) 83–93.

[28] H.J. Zimmermann, *Fuzzy Sets, Decision Making and Expert Systems* (Kluwer Academic Publishers, Boston, 1987).

[29] H.J. Zimmermann, *Fuzzy Set Theory and its Applications* (Kluwer Academic Publisher, Boston, 1991).