ELSEVIER

Fuzzy Sets and Systems 112 (2000) 117–125

FUZZY sets and systems

www.elsevier.com/locate/fss

On the optimization of fuzzy decision trees

Xizhao Wang[a,*], Bin Chen[b], Guoliang Qian[b], Feng Ye[b]

[a] *Department of Mathematics, Hebei University, Baoding 071002, Hebei, People's Republic of China*
[b] *Machine Learning Group, Department of Computer Science, Harbin Institute of Technology, Harbin 150001, People's Republic of China*

Received June 1997; received in revised form November 1997

Abstract

The induction of fuzzy decision trees is an important way of acquiring imprecise knowledge automatically. Fuzzy ID3 and its variants are popular and efficient methods of making fuzzy decision trees from a group of training examples. This paper points out the inherent defect of the likes of Fuzzy ID3, presents two optimization principles of fuzzy decision trees, proves that the algorithm complexity of constructing a kind of minimum fuzzy decision tree is NP-hard, and gives a new algorithm which is applied to three practical problems. The experimental results show that, with regard to the size of trees and the classification accuracy for unknown cases, the new algorithm is superior to the likes of Fuzzy ID3.

Keywords: Machine learning; Learning from examples; Knowledge acquisition and learning; Fuzzy decision trees; Complexity of fuzzy algorithms; NP-hardness

Notation

X	a given finite set
$F(X)$	the family of all fuzzy subsets defined on X
A, B, C, P, N	fuzzy subsets defined on X, i.e. values of attributes (or N_i, $A_i^{(k)}$, etc. fuzzy classification) i.e. nodes of fuzzy decision trees
$M(\cdot)$	the cardinality of a fuzzy subset
$\lvert\cdot\rvert$	the cardinality of a crisp subset
Ω_i, Γ	groups of fuzzy subsets, i.e., attributes
$p_i^{(k)}, q_i^{(k)}$	relative frequencies concerning some classes
$[\Omega = A]$	a branch of a decision tree
$f_j^i = P(\Omega = A_i \mid C_j)$	the conditional frequency
$Entr_i^{(k)}$	fuzzy entropy
$Ambig_i^{(k)}$	classification ambiguity

1. Introduction

Machine learning is the essential way to acquire intelligence for any computer system. Learning from examples, i.e. concepts acquisition, is one of the most important branches of machine learning. It has been generally regarded as the bottle-neck of expert system development.

* Corresponding author.
E-mail address: wangxz@hbu.edu.cn (X. Wang)

0165-0114/00/$ - see front matter
PII: S0165-0114(97)00386-2

The induction of decision trees is an efficient way of learning from examples [8]. Many methods have been developed for constructing decision trees [9] and these methods are very useful in building knowledge-based expert systems. Cognitive uncertainties, such as vagueness and ambiguity, have been incorporated into the knowledge induction process by using fuzzy decision trees [14]. Fuzzy ID3 algorithm and its variants [2, 4, 10, 12–14] are popular and efficient methods of making fuzzy decision trees. The fuzzy ID3 can generate fuzzy decision trees without much computation. It has the great matching speed and is especially suitable for large-scale learning problems.

Most of existing algorithms for constructing (fuzzy) decision trees focus on the selection of expanded attributes (e.g. [1, 4, 7, 8, 10, 13, 14]). They attempt to obtain a small-scale tree via the expanded attribute selection and to improve the classification accuracy for unknown cases. Generally, the smaller the scale of decision trees, the stronger their generalizing capability. It is possible that the reduction of decision tree scale results in the improvement of the classification accuracy for unknown cases. An important problem is whether or not there exists an exact algorithm for constructing the smallest-scale fuzzy decision tree.

Centering on this optimal problem, this paper discusses the representation of fuzzy decision trees and gives a comparison between fuzzy decision trees and crisp ones; proves that the algorithm complexity for constructing a kind of smallest-scale fuzzy decision tree is NP-hard; points out the inherent defect of the fuzzy ID3 and presents a new algorithm for constructing fuzzy decision trees; and applies the new algorithm to three practical problems and shows the improvement of accuracy (in comparison with the fuzzy ID3 algorithm).

2. Fuzzy decision trees

Consider a directed tree of which each edge links two nodes, the initial node and the terminal node. The former is called the fathernode of the latter while the latter is said to be the sonnode of the former. The node having not its fathernode is the root whereas the nodes having not any sonnodes are called leaves.

A decision tree is a kind of directed tree. The induction of decision trees is an important way of inductive learning. It has two key aspects, training and matching. The former is a process of constructing trees from a training set which is a collection of objects whose classes are known, while the latter is a process of judging classification for unknown cases. ID3 is a typical algorithm for generating decision trees [8].

A fuzzy decision tree is a generalization of the crisp case. Before giving the definition, we explain some symbols. Throughout this paper, X is a given finite set, $F(X)$ is the family of all fuzzy subsets defined on X, and the cardinality measure of a fuzzy subset, A, is defined by $M(A) = \sum_{x \in X} A(x)$ (the membership of a fuzzy subset is denoted by itself).

Definition 1. Let $\Omega_i \subset F(X)$ $(1 \leqslant i \leqslant m)$ be m given groups of fuzzy subsets, with the property $|\Omega_i| > 1$ ($|\cdot|$ denotes the cardinality of a crisp set). T is a directed tree satisfying

(a) each node of the tree belongs to $F(X)$,

(b) for each not_leaf, N, whose all sonnodes constitute a subset of $F(X)$ denoted by Γ, there exists i $(1 \leqslant i \leqslant m)$ such that $\Gamma = \Omega_i \cap N$, and

(c) each leaf corresponds to one or several values of classification decision.

Then, T is called a fuzzy decision tree. Each group of fuzzy subsets, Ω_i, corresponds to an attribute and each fuzzy subset corresponds to a value of the attribute.

Example 1. Consider Table 1 [14]. Each column corresponds to a fuzzy subset defined on $X = \{1, 2, 3, \dots, 16\}$, for instance, sunny $= 0.9/1 + 0.8/2 + 0.0/3 + \cdots + 1.0/16$. Four attributes are as follows.

$$\text{Outlook} = \{\text{Sunny, Cloudy, Rain}\} \subset F(X),$$

$$\text{Temperature} = \{\text{Hot, Mild, Cool}\} \subset F(X),$$

$$\text{Humidity} = \{\text{Humid, Normal}\} \subset F(X),$$

$$\text{Wind} = \{\text{Windy, Not\_Windy}\} \subset F(X).$$

In the last column, three symbols, V, S and W, denote three sports to play on weekends, Volleyball, Swimming and Weight_lifting, respectively. A fuzzy decision tree can be constructed by using the algorithm in [14] to train Table 1. It is shown as Fig. 1.

The general matching strategy of fuzzy decision trees is described as follows.

(a) Matching starts from the root and ends at a leaf along the branch of the maximum membership. (b) If the maximum membership at the node is not unique, matching proceeds along several branches. (c) The decision with maximum degree of truth is assigned to the matching result.

Example 2. Consider two examples remaining to be classified e1 = (0.0, 0.6, 0.4; 0.3, 0.7, 0.0; 0.5, 0.5; 0.5, 0.5) and e2 = (0.9, 0.1, 0.0; 0.8, 0.2, 0.0; 0.5, 0.5; 0.8, 0.2). The process of matching in the fuzzy decision tree (Fig. 1) is shown in Fig. 2.

Table 1
A small training set

No.	Outlook			Temperature			Humidity		Humid Normal		Class		
	Sunny	Cloudy	Rain	Hot	Mild	Cool	Humid	Normal	Windy	Not_windy	V	S	W
1	0.9	0.1	0.0	1.0	0.0	0.0	0.8	0.2	0.4	0.6	0.0	0.8	0.2
2	0.8	0.2	0.0	0.6	0.4	0.0	0.0	1.0	0.0	1.0	1.0	0.7	0.0
3	0.0	0.7	0.3	0.8	0.2	0.0	0.1	0.9	0.2	0.8	0.3	0.6	0.1
4	0.2	0.7	0.1	0.3	0.7	0.0	0.2	0.8	0.3	0.7	0.9	0.1	0.0
5	0.0	0.1	0.9	0.7	0.3	0.0	0.5	0.5	0.5	0.5	0.0	0.0	1.0
6	0.0	0.7	0.3	0.0	0.3	0.7	0.7	0.3	0.4	0.6	0.2	0.0	0.8
7	0.0	0.3	0.7	0.0	0.0	1.0	0.0	1.0	0.1	0.9	0.0	0.0	1.0
8	0.0	1.0	0.0	0.0	0.2	0.8	0.2	0.8	0.0	1.0	0.7	0.0	0.3
9	1.0	0.0	0.0	1.0	0.0	0.0	0.6	0.4	0.7	0.3	0.2	0.8	0.0
10	0.9	0.1	0.0	0.0	0.3	0.7	0.0	1.0	0.9	0.1	0.0	0.3	0.7
11	0.7	0.3	0.0	1.0	0.0	0.0	1.0	0.0	0.2	0.8	0.4	0.7	0.0
12	0.2	0.6	0.2	0.0	1.0	0.0	0.3	0.7	0.3	0.7	0.7	0.2	0.1
13	0.9	0.1	0.0	0.2	0.8	0.0	0.1	0.9	1.0	0.0	0.0	0.0	1.0
14	0.0	0.9	0.1	0.0	0.9	0.1	0.1	0.9	0.7	0.3	0.0	0.0	1.0
15	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.8	0.2	0.0	0.0	1.0
16	1.0	0.0	0.0	0.5	0.5	0.0	0.0	1.0	0.0	1.0	0.8	0.6	0.0

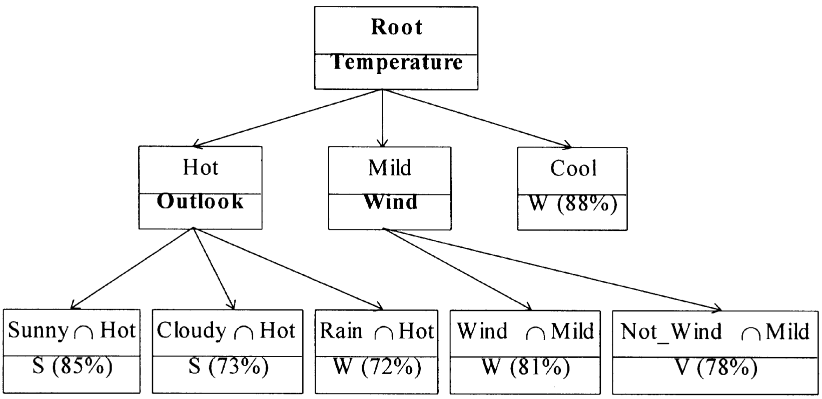

Fig. 1. A fuzzy decision tree generated by training Table 1. (The percentage attached to each decision is the degree of truth on the decision.)

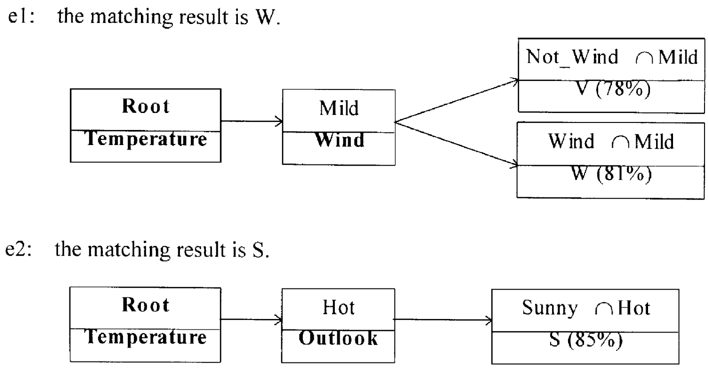

Fig. 2. The matching process of two examples. e1: the matching result is W. e2: the matching result is S.

Table 2
A comparison between the fuzzy decision tree and the crisp case

Crisp decision tree	Fuzzy decision tree
Nodes are crisp subsets of X	Nodes are fuzzy subsets of X
If N is not a leaf and $\{N_i\}$ is the family of all sonnodes of N, then $\bigcup_i N_i = N$	If N is not a leaf and $\{N_i\}$ is the family of all sonnodes of N, then $\bigcup_i N_i \subset N$
A path from the root to a leaf corresponds to a production rule	A path from the root to a leaf corresponds to a fuzzy rule with some degree of truth
An example remaining to be classified matches only one path in the tree	An example remaining to be classified can match several paths in the tree
The intersection of subnodes located on the same layer is empty	The intersection of subnodes located on the same layer can be nonempty

The fuzzy decision tree, regarded as a generalization of the crisp case, is more robust in tolerating imprecise information. Compared with the crisp case, the fuzzy decision tree has the characteristics listed in Table 2.

There are many methods for representing fuzzy decision trees. In essential, the representation in this paper is the same as in the article [14] where the fuzzy decision tree is regarded as fuzzy partitioning.

3. Optimization of fuzzy decision trees

There are two key points in the process of constructing fuzzy decision trees. One is the selection of expanded attributes. They are such attributes that according to values of attributes (which are fuzzy subsets) trees are expanded at the nodes considered i.e. sonnodes of the nodes are generated. The other is the judgment on leaves. Nodes are usually regarded as leaves if the relative frequency of one class is greater than or equal to a given threshold value.

Learning algorithms typically use heuristics to guide their search. A general learning algorithm for generating fuzzy decision trees can be described as follows.

Consider the whole training set which is regarded as the first candidate node.
WHILE there exist candidate nodes
DO select one using the search strategy; if the selected one is not a leaf, then generate its sonnodes by selecting the expanded attribute using a heuristic. These sonnodes are regarded as new candidate nodes.

From existing references, two powerful heuristics guiding the selection of expanded attributes in the fuzzy decision tree generation can be found. One, called Fuzzy-ID3, is based on minimum fuzzy entropy [2, 4, 10, 13] whereas the other is based on the reduction of classification ambiguity [14]. The latter is a variant of the former. These two heuristics are briefly described as follows.

Let there be N training examples and n attributes $A^{(1)}, \ldots, A^{(n)}$. For each k ($1 \leqslant k \leqslant n$), the attribute $A^{(k)}$ takes m_k values of fuzzy subsets, $A_1^{(k)}, \ldots, A_{m_k}^{(k)}$. For simplicity, the fuzzy classification is considered to be two fuzzy subsets, denoted by P and N respectively. For each attribute value (fuzzy subset), $A_i^{(k)}$ ($1 \leqslant k \leqslant n, 1 \leqslant i \leqslant m_k$), its relative frequencies concerning P and N are

$$p_i^{(k)} = M(A_i^{(k)} \cap P)/M(A_i^{(k)})$$

and

$$q_i^{(k)} = M(A_i^{(k)} \cap N)/M(A_i^{(k)}),$$

respectively. The relative frequency is regarded as the degree of truth of a fuzzy rule in [14].

Heuristic (1). Fuzzy ID3 based on the minimum fuzzy entropy.

Select such an integer k_0 (the k_0th attribute) that $E_{k_0} = \mathrm{Min}_{1 \leqslant k \leqslant n} E_k$, where

$$E_k = \sum_{i=1}^{m_k} \left(M(A_i^{(k)}) \middle/ \sum_{j=1}^{m_k} M(A_j^{(k)}) \right) Entr_i^{(k)},$$

$$k = 1, 2, \ldots, n;$$

$$Entr_i^{(k)} = -\, p_i^{(k)} \log_2 p_i^{(k)} - q_i^{(k)} \log_2 q_i^{(k)}$$

denotes the fuzzy entropy of classification.

Heuristic (2). A variant of Fuzzy ID3 based on the minimum classification ambiguity.

Select such an integer k_0 (the k_0th attribute) that $G_{k_0} = \mathrm{Min}_{1 \leqslant k \leqslant n} G_k$, where

$$G_k = \sum_{i=1}^{m_k} \left(M(A_i^{(k)}) \middle/ \sum_{j=1}^{m_k} M(A_j^{(k)}) \right) Ambig_i^{(k)},$$

$$k = 1, 2, \ldots, n;$$

$$Ambig_i^{(k)} = \mathrm{Min}(p_i^{(k)}, q_i^{(k)})/\mathrm{Max}(p_i^{(k)}, q_i^{(k)})$$

denotes the ambiguity of classification.

The heuristics (1) and (2) can be easily extended to the case of fuzzy classification with more than two fuzzy subsets. The heuristic (2) has an option of significant level which will affect the generation of fuzzy decision trees. A fuzzy decision tree, as shown in Fig. 2, is generated by using the heuristic (2) where the significant level is taken to be 0.5. The same fuzzy decision tree can be generated by using the heuristic (1). It is just a coincidence. Generally, different heuristics will result in different trees. Intuitively, both the weighted average of fuzzy entropies and the weighted average of classification ambiguities will decrease when

$$\mathrm{Max}(p_i^{(k)}, q_i^{(k)}) \to 1 \quad \text{or} \quad \mathrm{Min}(p_i^{(k)}, q_i^{(k)}) \to 0.$$

In the process of generating fuzzy decision trees, both heuristic (1) and heuristic (2) attempt to reduce the average depth of trees, i.e. on an average, to generate leaves as soon as possible.

One of main objectives of fuzzy decision tree induction is to generate a tree with high accuracy of classification for unknown cases. Which factors influence the accuracy? Experimental results [1] show, at least, that the selection of expanded attributes is an important factor. Most existing researches on fuzzy (crisp) decision trees are focused on this selection (e.g. [1, 4, 7, 8, 10, 13, 14]). In the following, we analyze this problem from the optimization angle.

Essentially, the accuracy of the classification for unknown cases depends mainly on two aspects, the size of decision trees and the representativeness of training data. This paper considers only the former. For two decision trees generated by training the same data, one prefers to select the small-scale tree for classifying unknown cases. There are several norms for measuring the scale of decision trees, that is, the complexity can be defined in various ways, but common measures include the total number of leaves and the average depth of decision trees. One naturally hopes that the total number is as few as possible and the average depth is as little as possible respectively. Based on this viewpoint, two optimization problems concerning fuzzy decision trees are proposed as follows.

(A) Look for a fuzzy decision tree with the minimum total number of leaves.

(B) Look for a fuzzy decision tree with the minimum average depth of leaves.

The following theorem gives us the computation complexity for the problem (A). (We guess that the problem (B) is also NP hard, but cannot give the proof.) The proof of the theorem and relative explanation are given in the last part of this section.

Theorem. *The problem* (A) *is NP hard.*

A computational problem is said to be a P-problem if there exists such an algorithm that the exact solution can be obtained within polynomial time, if not, it is called NP-hard. So far, only the heuristic algorithm concerning the approximate solution can be given for the NP-hard. It plays a key role in computation complexity theory that the algorithm for obtaining the exact solution is given or the problem is proved to be NP-hard. Details of NP-hard problems can be found in [3, 11].

It is unrealistic to find the exact algorithm for the problem (A) because it is NP hard. Common methods are to use heuristics to look for approximately optimal solutions. Heuristics used in the likes of fuzzy ID3 attempt to reduce only the average depth of leaves (e.g. heuristics (1) and (2) mentioned above), but neglect discussing the number of leaves. This is the inherent defect of the likes of fuzzy ID3 algorithm. Our research shows that just the number of leaves plays a key role in improving the accuracy of classification for unknown cases. Drawing inspiration from these two optimization problems, we design a new induction of fuzzy decision trees, called MB algorithm, in the following section.

Proof of the theorem. Let A and B be two problems. If A is NP hard and A can be reduced to B within polynomial time, then B is also NP hard. The following problem of optimal set cover has been proved to be NP hard in [5].

Problem of optimal set cover: Let $T = \{1, 2, \ldots, m\}$, $F = \{S_1, \ldots, S_p\}$, $S_i \subset T$ and $\bigcup_{S\in F} S = T$. If G is such a subfamily that $G \subset F$, $\bigcup_{S\in G} S = T$ and $|G| =$ minimum, then G is called an optimal set cover of T, where $|G|$ denotes the cardinality of G.

We will accomplish the proof via reducing the problem of optimal set cover to the problem of decision tree with minimum number of leaves. Without losing generality, we illustrate how to construct a transformation to implement the reducing process by using a concrete example.

Let $T = \{1, 2, 3, \ldots, 7\}$, $F = \{S_1, \ldots, S_6\} = \{\{1, 4, 5, 7\}, \{3, 4\}, \{2, 5, 7\}, \{1, 2, 6\}, \{1, 3, 7\}, \{3, 5, 6\}\}$. It is obvious that F is a set cover of T, i.e. $\bigcup_{S\in F} S = T$. Constructing a characteristic table (Table 3) of this set cover, we regard Table 3 as 7 positive examples and $e = (0, 0, 0, 0, 0, 0)$ as the uniquely negative example. In Table 3, F_i corresponds to an attribute $S_i = [F_i \neq 0]$ holds for each i ($1 \leqslant i \leqslant 6$).

Each set cover of T corresponds to a fuzzy decision tree. For example, $\{S_1, S_2, S_4\}$ is a set cover of T and the corresponding fuzzy decision tree is shown as Fig. 3. The corresponding tree is fuzzy since the intersection of subnodes located on the same layer is not empty. It is easy to see that the number of subsets of the set cover = the number of leaves of the corresponding tree + 1.

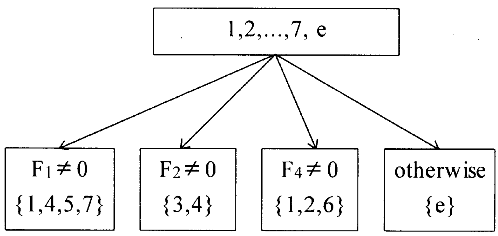

Fig. 3. A fuzzy decision tree corresponding to the set cover $\{S_1, S_2, S_4\}$.

Table 3
A characteristic table of the set cover F

No.	F_1	F_2	F_3	F_4	F_5	F_6
1	1	0	0	1	1	0
2	0	0	1	1	0	0
3	0	1	0	0	1	1
4	1	1	0	0	0	0
5	1	0	1	0	0	1
6	0	0	0	1	0	1
7	1	0	1	0	1	0

Now, we have to convert the problem of optimal set cover into the problem of fuzzy decision tree with minimum number of leaves, which is just the desired result. The proof is completed.

4. A new algorithm for constructing fuzzy decision trees

In the process of expanding a Not_leaf node on the fuzzy decision tree, the new algorithm first selects the expanded attribute using heuristic (1) or (2) mentioned previously, and then uses fuzzy value clustering to merge branches which the Not_leaf node sends out. Merging branches is essential for the new algorithm.

The aim of selecting the expanded attribute via using heuristic (1) or (2) mentioned in Section 3 is an attempt to reduce the average distance from the current node to its childnodes, i.e. attempt to satisfy the optimization principle (B) mentioned in Section 3. It guarantees that the maximum classification information can be obtained when an unknown example is tested on the current Not_leaf node. The aim of merging branches is an attempt to guarantee that the number of branches which the current Not_leaf node sends out are as few as possible. That is, merging branches attempts to satisfy the optimization principle (A) mentioned in Section 3. Key points of the new algorithm are listed as follows.

Let Ω be the expanded attribute at the current Not_leaf node and Ω has n values of fuzzy subsets defined on X, $\{A_1, A_2, \dots, A_n\}$. The classification is described by m fuzzy subsets, $\{C_1, C_2, \dots, C_m\}$.

(a) For fixed i ($1 \leqslant i \leqslant n$), compute the conditional frequency

$$f_j^i = P(\Omega = A_i \,|\, C_j) \quad \text{for } j = 1, 2, \dots, m,$$

where $P(\Omega = A_i \,|\, C_j) = M(A_i \cap C_j)/M(C_j)$.

(b) For fixed i ($1 \leqslant i \leqslant n$), select an integer $k = k(i)$ (k depends on i) such that

$$f_k^i = \operatorname*{Max}_{1 \leqslant j \leqslant m} f_j^i.$$

Then put the label, "$k(i)$-*class*", on the branch $[\Omega = A_i]$.

(c) Cluster according to the following rule:

if $k(i_1)$-*class* = $k(i_2)$-*class*
then the two branches, $[\Omega = A_{i_1}]$ and $[\Omega = A_{i_2}]$, merge into one branch.

After clustering, values of the attribute Ω are divided into L groups. Generally, $L \leqslant n$.

(d) Expand the current Not_leaf node, each group corresponds to a new branch (i.e., a new sonnode).

Compared with the original heuristic, the new algorithm makes the number of branches which the current Not_leaf node sends out to decrease. We call the new algorithm Merging Branches algorithm, in short, MB algorithm.

Example 3. Using MB algorithm where the selec tion of the expanded attribute is based on heuristic (1) to train Table 1, we obtain the fuzzy decision tree which is shown as Fig. 4.

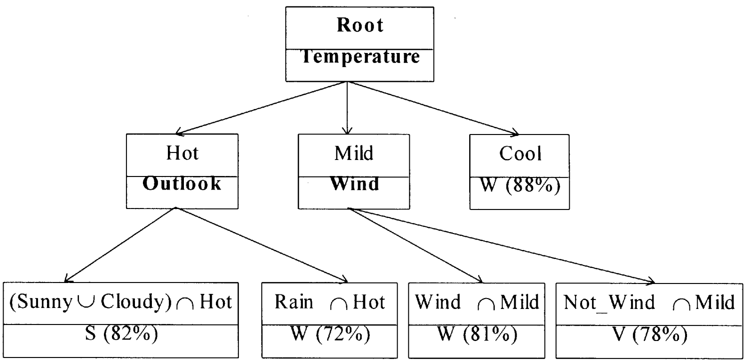

Fig. 4. A fuzzy decision tree constructed by using the new algorithm.

Table 4
An experimental comparison between the Fuzzy ID3 and the MB

Problem	Algorithm	Number of leaves	Test accuracy	Training time (s)
Sleep	Fuzzy ID3	288	88%	18
	MB	206	95%	32
RHCC	Fuzzy ID3	602	72%	45
	MB	463	81%	71
Diagnosis	Fuzzy ID3	12	86%	—
	MB	8	96%	—

5. Analysis and comparison of experimental results

In the Pentium-586 microcomputer, we implement the Fuzzy ID3 algorithm and the MB algorithm using C++ Language and apply them to three practical problems which are as follows.

The first problem is the automatic knowledge acquisition about sleep states which is provided by Illinois University [6]. It is a crisp case which consists of a group of typical data concerning learning from examples. It provides 1236 examples which are divided into 6 classes. Each example has 11 attributes. We randomly select a training set of 1000 examples and construct two decision trees using ID3 and MB, respectively. The experimental results of testing the remaining 236 examples are shown in Table 4.

The second problem is the recognition of handwritten Chinese characters (RHCC) which possesses much fuzziness since their deformation is very serious. We select 100 Chinese characters used frequently, ask 30 students to write them, and then obtain 3000 examples. These examples regarded as images are inputted by a scanner. After extracting 12 fuzzy features (attributes), we select a training set of 2000 examples and obtain the experimental results shown in Table 4.

The third problem is the fault diagnosis of turbine generators [15]. It is a learning problem of continuous-valued attributes. It provides 40 examples divided into 4 types of faults. Each example has 8 continuous-valued attributes. After fuzzifying a training set of 30 examples, we obtain the experimental results shown in Table 4.

The experimental results in Table 4 show that the size of each tree and the test accuracy of the MB algorithm are superior to that of the Fuzzy ID3 algorithm. But the training speed of the former is slightly less than that of the latter and the selection of expanded attributes in the MB algorithm can be repeated. It can be seen that the MB algorithm is a reasonable compromise between the training speed and the test accuracy.

6. Conclusions

This paper discusses the optimization of fuzzy decision trees and has the following main results:

(1) The computation complexity of making the minimum fuzzy decision tree is NP hard.

(2) The likes of Fuzzy ID3 algorithm have the inherent defect and, to some degree, the new algorithm presented in this paper remedies this defect.

(3) With regard to the number of leaves and the classification accuracy for unknown cases, the new algorithm is superior to the fuzzy ID3.

(4) The computation complexity of the optimization problem (B) mentioned in Section 3 remains to be studied further.

References

[1] W. Buntine, T. Niblett, A further comparison of splitting rules for decision-tree induction, Machine Learning 8 (1992) 75–85.

[2] K.J. Cios, L.M. Sztandera, Continuous ID3 algorithm with fuzzy entropy measures, Proc. IEEE Internat. Conf. on Fuzzy Systems, San Diego, CA, March 1992, pp. 469–476.

[3] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Printed in the United States of America, 1979.

[4] H. Ichihashi et al., Neuro-Fuzzy ID3, Fuzzy Sets and Systems 81 (1996) 151–167.
[5] D.S. Johnson, Approximate algorithm for combinational problems, J. Comput. Systems Sci. 9(3) (1973).
[6] R.S. Michalski, I. Mozetic and J.R. Hong, The multipurpose incremental learning system AQ15, Proc. 5th AAAI (1986) 1041–1045.
[7] J. Mingers, An empirical comparison of selection measures for decision-tree induction, Machine Learning 3 (1989) 319–342.
[8] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[9] S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE Trans. System Man Cybernet. 21 (1991) 660–674.
[10] T. Tani, M. Sakoda, Fuzzy Modeling by ID3 algorithm and its application to prediction, Proc. IEEE Internat. Conf. on Fuzzy Systems, San Diego, CA, March 1992, pp. 923–930.
[11] J.F. Traub, Algorithms and Complexity: New Directions and Recent Results, Academic Press, New York, 1976.
[12] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Trans. System Man Cybernet. 22 (1992) 1414–1427.
[13] R.Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, Proc. 2nd Internat. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan, 17–22 July 1992, pp. 265–268.
[14] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.
[15] L. Zhansheng, Study on Diagnostic Expert Systems and its Knowledge acquisition for Rotating Machinery, Ph.D. Dissertation, Harbin Institute of Technology, 1995.