ELSEVIER

Fuzzy Sets and Systems 130 (2002) 87–99

FUZZY sets and systems

www.elsevier.com/locate/fss

Fuzzy classification trees for data analysis

I-Jen Chiang[a, *], Jane Yung-jen Hsu[b]

[a] *Department of Medical Informatics, Taipei Medical University, Taipei, Taiwan 105, ROC*
[b] *Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan 100, ROC*

Received 21 March 1997; received in revised form 2 October 2001; accepted 10 October 2001

Abstract

Overly generalized predictions are a serious problem in concept classification. In particular, the boundaries among classes are not always clearly defined. For example, there are usually uncertainties in diagnoses based on data from biochemical laboratory examinations. Such uncertainties make the prediction be more difficult than noise-free data. To avoid such problems, the idea of *fuzzy classification* is proposed. This paper presents the basic definition of fuzzy classification trees along with their construction algorithm. *Fuzzy classification trees* is a new model that integrates the fuzzy classifiers with decision trees, that can work well in classifying the data with noise. Instead of determining a single class for any given instance, fuzzy classification predicts the degree of *possibility* for every class.

Some empirical results the dataset from UCI Repository are given for comparing FCT and C4.5. Generally speaking, FCT can obtain better results than C4.5.

Keywords: Artificial intelligence; Decision making; Classifications; Information theory; Decision trees; Tree classifiers

1. Introduction

Discovering regularities in complex data is an important research topic. Many problems in medical, astronomical, and financial applications involve a large amount of data that need to be classified. Classification can be thought as the base of ability to knowledge acquisition [25]. Current classification techniques, e.g. decision trees [13,22–24,30,33] work well for pattern recognition and process control. Unfortunately, while we are considering the problem of uncertainties and noise, the data would be very difficult to be clearly classified. According to the following example, it can be identified why those approaches are failed to classify the data.

Example 1. Table 1 lists the instances for a classification task to decide whether a patient is at risk of having a stroke. The attributes are systolic and diastolic arterial blood pressures.

Figs. 1 and 2 show the decision tree generated by C4.5/ID3 [35]. A new instance with blood pressures of `systolic` = 154 and `diastolic` = 74 will be classified as a stroke patient.

Such a conclusion may be incorrect. As we all know, the risk of stroke for people with normal diastolic arterial blood pressures is usually not high. It is most likely that the patient will not suffer from a stroke. However, in rare cases, the abnormal systolic arterial blood pressure may be caused

* Corresponding author. 3F, 8-1 Tai-An Street, Taipei 100, Taiwan.
E-mail addresses: chiang@robot.csie.ntu.edu.tw (I-J. Chiang), yjhsu@csie.ntu.edu.tw (J.Y.-j. Hsu).

0165-0114/02/$ - see front matter
PII: S0165-0114(01)00212-3

Table 1
A training set about *stroke* patients

No.	Systolic	Diastolic	Class
1	170	75	Normal
2	180	67	Normal
3	170	95	Stroke
4	181	72	Stroke
5	194	56	Normal
6	195	54	Stroke
7	169	82	Stroke
8	144	90	Normal

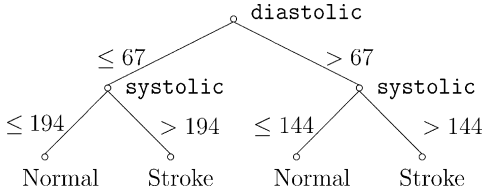

Fig. 1. A decision tree for Example 1.

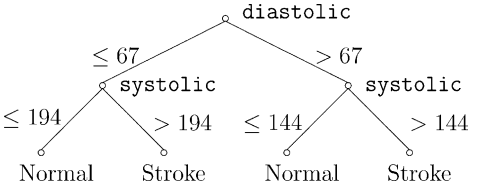

Fig. 2. A decision tree for Example 1.

by a chronic condition which leads to a stroke eventually.

As a result, a more reasonable answer from the classification system should present all probable conclusions, each of which is associated with a degree of possibility. Therefore, fuzzy classifications have proposed by Hsu and Chiang [9,14].

In this paper, we will present to use this approach to classification in domains with such vague conclusions. Some related work is given in Section 2. The definitions of *fuzzy classification trees* are presented in Section 3. The attribute selection measures are defined in Section 4. Section 5 describes the basic algorithm for constructing a FCT from a data set. Section 6 shows the empirical results compared FCT with C4.5 on some UCI repository data sets. The advantages and limitations of fuzzy classification are discussed in Section 7, followed by the conclusion.

2. Related work

When the number of variable to describe a process is not large, it models the process by (1) dividing the whole space into several subspaces, (2) representing each subspace by a simple linear function, and (3) interpolating several subspaces continuously. When a system is very complex, it is necessary to extract the relevant variables in the premises of fuzzy models. Sugeno and Kang [42] proposed to use a mathematical programming method to dealing with this program. A large amount of calculation to identify premise parameters is unavoiding.

Many methods have been developed for constructing decision trees from collections of examples. Although these methods are useful in building knowledge based expert system, they often suffer from inadequately or improperly expressing and handling the vagueness and ambiguity associated with human thinking and perception [47]. Even by the Quinlan's work [31], the types of uncertainties are not to be probabilistic, appearing as randomness or noise. Pedrycz and Sosnowski [26] pointed out that the concept of fuzzy granulation realized via context-based clustering is aimed at the discretization process. For the sake of vagueness, fuzzy decision trees are issued.

Hunt et al. [16] proposed *Concept learning system* to construct a decision tree that attempts to minimize the score of classifying chess endgames. Quinlan modified CLS and proposed the ID3 algorithm [27,28]. ID3 represents acquired knowledge in the form of decision trees. An internal node

of a tree specifies a test of an attribute, with each outgoing branch corresponding to a possible result of this test. Leaf nodes represent the class to be assigned to an instance. Quinlan abandoned the cost-driven lookahead of CLS with an information-driven evaluation function to solve the Michie's challenge for chess endgames. The evaluation function, called *entropy measure*, is used to decide the pattern-based features from the chess position. In order to classify an instance, ID3 starts at the root of the tree, evaluates the test (attribute), and takes the branch appropriate to the outcome. Only a subset of the attributes may be encountered on a particular path from the root of a decision tree to a leaf.

Quinlan further refined ID3 [29–32,34]. The concept of processing noisy data and unknown attribute values from ASSISTANT [19] is embedded into ID3 and its successor, C4 [31,34]. During induction, a best attribute tests selected by an information-based measure of all possible attributes is used to 'spawn' a leaf node in the tree. A numerical attribute is discretized by a cut point and is considered to be a binary categorical data. This makes the numerical attributes process as categorical attribute. Therefore, all the attributes are taken to be categorical. However, the clear cut makes the data too straightforward to be partitioned. The minority of the overlapped data has been misclassified. Overfitting of decision trees can thus be avoided by halting the tree growth when no more significant information can be gained. Stopping recursively building the tree by irrelevant attribute tests, such as the information gain of any tested attribute exceeds a threshold or χ^2 test for stochastic independent, is considered in ID3 and C4 for noisy data processing. C4.5 [35] is the summary of Quinlan's ID3 algorithms.

2.1. Binary fuzzy decision trees

Fuzzy decision trees were first mentioned by Chang and Pavlidis [7]. In their paper, the fuzzy decision tree is defined to be a binary tree in which each nonterminal node contains four fields including one decision attribute and tree links. The links of a node are the pointers to its parent, and the pointers to its left and right children, respectively. Instead of using the efficient top-down search but less correct than the inefficient bottom-up search for parsing the tree, they presented the branch-bound-backtrack algorithm. This tree search method is belongs to the family of branch-and-bound methods. This paper presented the structure of fuzzy decision trees, a search method and the relationship between decision trees and fuzzy decision trees. The paper did not address how to construct the fuzzy decision trees.

Wang and Suen [44] used fuzzy regions to cover the Bayes decision regions. They applied their work to 3200 Chinese characters recognition. Since the error accumulation form classification on decision trees can be very harmful when the number of classes is very large. They extended the regions with a prior probability to be fuzzy regions. The feature selection is based on the attribute whose minimum Mahalanobis distance is the maximum. The decision is evaluated by a heuristic function based on the membership functions. Fuzzy logic search is useful to find all possible correct classes, and the similarity measures are used to determine the most probable class. Global training is applied to expand the decision tree in order to enhance the recognition rate. That provides a lot of flexibility, and reduces the error accumulation.

2.2. Post-fuzzification

Cios and Sztandera [10] used a continuous ID3 algorithm convert a decision tree into a layer of a feedforward neural network. A neuron with a sigmoid function can be view as a hyperplane with fuzzy boundary. Kosko's fuzzy entropy is used to measure the fuzziness of classification by the neuron. The nodes within the hidden layer are generated until the fuzzy entropy is reduced to zero.

Tani et al. used ID3 to obtain the IF-THEN rule, and then used multiple regression analysis to fuzzify the premise of each rule. The premise of each rule is a conjunction of linguistic terms with associated membership functions. The membership function is used to determine the boundary of the fuzzy sets.

Maher and St. Clair [21] presented UR-ID3 to combine uncertain reasoning with the rule sets produced by ID3. UR-ID3 is a post-fuzzified method. After the ID3 decision tree is constructed, triangularly shaped membership functions is used to each of the decision values on the branches and attribute values. The classification of a test sample is done by the corresponding set of support intervals for each possible classification. Chi and Yan [8] following their

approach, converted the ID3 rules to fuzzy rules. To measure the degree of each IF-part to be satisfied and to measure the degree of overall antecedent conditions to be satisfied, suitable membership functions is proposed in their method. An defuzzification method to determine the output for each test input is generated from a two-layer perception. Using the same training sample which generated the fuzzy rules, a two-layer perceptron is trained to optimize the connection weights by minimizing a cost function.

Hsu et al. [15] used ID3 to generate the fuzzy control rules for mobile robot control. The rules are induced by ID3 from a collection of training data. This data is a combination of sensors' value and robot's actions. Like UR-ID3, a post-fuzzification is applied to the generated rules. The fuzzy rules are represented by a neural network architecture. The gradient-descent approach is used to turn the membership function of each linguistic variable while performing an on-line training.

Suárez and Lutsko [41] generated partial membership in the nodes of a CART decision tree by incorporating features of connectionist methods. After a decision tree has been generated, a reformulation of the tree construction algorithm in terms of fuzzy degrees of membership makes it possible to employ analytic tools in the construction decision trees that are globally optimal.

Boyen and Wehenkel [2] proposed to use neural networks, multilayer perceptrons, to implemented to generate a fuzzy decision tree for the power system security assessment.

2.3. Pre-fuzzification

Weber presented Fuzzy-ID3 [45,46]. No fuzziness is involved with categorical attributes. Numerical attribute values are fuzzified into linguistic terms before induction. The probability of fuzzy event is used to replace the probability of crisp value for numerical attribute values. The fuzzy entropy [20] is used to measure the disorder of the fuzzified data. According to the fuzzy entropy of each attribute, the most suitable attribute is selected for branching. The branching form decision node seems somehow overlapping, but not being treated as fuzzy partitioning.

Yuan and Shaw [47] categorize the uncertainties into two categories: statistical and cognitive. There are two components in the cognitive uncertainty: *fuzziness* and *ambiguity*. All the previous work, as Yuan and Shaw pointed, loses to concern about the ambiguity uncertainty. Yuan and Shaw proposed an induction learning algorithm for fuzzy decision trees. They focused on incorporating cognitive uncertainties into knowledge induction process for classification. Each attribute value is first fuzzified into each linguistic term in a set and is specified by a membership value between 0 and 1. Once the fuzzy sets are introduced, the cognitive uncertainties can therefore be measured by fuzziness (vagueness) measures and ambiguity measures. The fuzzy entropy [20] and the nonspecificity measure are used to measure the vagueness and ambiguity uncertainties. Based on ID3, the nonspecificity measure is used as the goodness of split in constructing the fuzzy decision tree. The fuzzy decision tree is constructed by reducing the classification ambiguity. Form the rules induced by the fuzzy decision algorithm, the most possible class can be found.

Janickow [18] proposed to use *Exemplar-based* learning associated with fuzzy attributes to build up the fuzzy decision tree. Special examples are selected or generated from data to be used with a proximity measure, which is represented as membership function. Following the ID3 constructing algorithms, Janickow's algorithm explands the tree by using the fuzzy operations on the fuzzy set at each node.

However, no matter what the fuzzy decision tree methods are, they all unavoid two phases processing to generate the decision rules. They either prefuzzify the data according to domain knowledge or postfuzzify the decision rules generated by the decision tree methods by some tuning methods. They do not concern the distribution of the data that can make an improper classifications.

3. Definitions

This section introduces the concept of *fuzzy classification* and its basic definitions [14].

Classification problems are concerned with assigning classes to instances. Each instance can be described in terms of a set of attribute values, which are used as the basis for classification. Therefore, given an arbitrary set of data, the most important issue is to identify their key attributes.

Consider an ordered set of attributes $A=(a_1,a_2,\ldots,a_n)$ for instance description. For example, to decide whether one should play golf depends on attributes $\{\texttt{Outlook},\texttt{Temperature},\ldots\}$. An *attribute value vector* $\mathbf{x}=\langle x_1,x_2,\ldots,x_n\rangle$ consists of the value x_i for the corresponding attribute a_i. Each attribute may take either *ordered* or *categorical* values. Ordered values are typically numerical, either discrete or continuous, while categorical values are symbolic. The attribute value vector for the golf example may look like $\langle\texttt{sunny},85°\text{F},\ldots\rangle$, in which attribute `Outlook` takes a symbol as value and attribute `Temperature` is numerical. As another example, in a study on physical examinations performed at the National Taiwan University, there were more than 400 attributes associated with the examinations. The first attribute was sex, which had a symbolic value of either male or female; the second attribute, age, had a discrete value ranging from 0 to 120; the third attribute, height, was defined over a continuous range from 0 to 200.

3.1. Classifications

A *classification problem* is defined as a pair $(\mathscr{X},\mathscr{C})$, where $\mathscr{X}$, called the *instance space*, is the collection of all possible instances, and $\mathscr{C}=\{C_1,C_2,\ldots,C_n\}$ is the set of all possible classes. A classifier maps each instance into a class. Formally,

Definition 1. A *classifier* D for a given classification problem $(\mathscr{X},\mathscr{C})$, defines a total function

$$D:\mathscr{X}\to\mathscr{C}.$$

Every instance $\mathbf{x}\in\mathscr{X}$ is classified by the decision function into a single class $D(\mathbf{x})$.

The goal of classification is to find a classifier D that *correctly* classifies the given set of instances. Let $\text{Class}(\mathbf{x})$ denote the actual class of an instance $\mathbf{x}\in\mathscr{X}$. A instance $\mathbf{x}$ is said to be *misclassified*, if

$$D(\mathbf{x})\neq\text{Class}(\mathbf{x}).$$

In general, a classifier is considered to be better than another if it has a lower *misclassification rate*, which is defined as the probability that any instance $\mathbf{x}\in\mathscr{X}$ is misclassified.

A class of classifiers called *tree classifiers* are of particular interests in solving practical classification problems. A tree classifier determines the class of an instance based on a sequence of tests on its attributes. A global decision is reached via a series of local decisions that constitute a path through a tree structure.

Decision trees provide a straightforward implementation of tree classifiers [37,38]. Each terminal node (or leaf) in a decision tree is labeled with a class together with a set of instances, while each nonterminal node is labeled with a test on certain attribute value(s). Each test defines the branches from its associated node, and each branch is labeled with an attribute value (or a range of values). An instance follows a unique choice of branch from a node according to its attribute values.

3.2. Fuzzy classifications

Although tree classifiers are generally efficient, they have serious problems in dealing with elaborate real-valued attributes [12,11]. As Example 1 in the previous section shows, standard decision trees cannot handle multiple instances with overlapping attribute values that belong to different classes. To overcome such difficulties, the idea of *fuzzy classification* is proposed below.

Definition 2. Given a *fuzzy classifier* $\mathbf{F}$ for a given classification problem $(\mathscr{X},\mathscr{C})$ defines a total function

$$\mathbf{F}:\mathscr{X}\to\{\langle p_1,\ldots,p_n\rangle\,|\,p_i\in[0,1]\},$$

where p_i is the *possibility* that a given instance $\mathbf{x}$ belongs to class C_i.

For ease of presentation, the function $\mathbf{F}$ is sometimes represented as a vector of functions

$$\langle\wp_1,\wp_2,\ldots,\wp_n\rangle,$$

where $\wp_i$ is a possibility function $\mathscr{X}\to[0,1]$. For any given instance $\mathbf{x}$, the relation $\wp_i(\mathbf{x})>\wp_j(\mathbf{x})$ indicates that it is more likely for the instance $\mathbf{x}$ to be in class C_i.

A fuzzy classifier can be readily implemented by a tree structure. This section presents the basic definitions of *fuzzy classification trees* (FCTs). To facilitate discussions in the rest of the paper, we define a labeling scheme that assigns a unique label for each node.

Definition 3. Given an *FCT*, each node n in the tree $\mathscr{T}$ is given a label:

$$\text{Label}(n) = \begin{cases} 1 & \text{if } n \text{ is the root,} \\ \text{Label}(n').i & \text{if } n \text{ is the } i\text{th} \\ & \text{child of node } n'., \end{cases}$$

where . is the concatenation operator.

Let $\mathscr{L}$ be the set of all labels, N_L denote the node labeled by $L \in \mathscr{L}$, and B_L denote the branch leading into node N_L. The label for the parent node of N_L can be easily obtained by removing the last integer from label L, and the result is denoted by $\hat{L}$. Each nonterminal node in the tree is associated with a test, and the resulting branches, $B_{L.i}$, is associated with a membership function

$$\mu_{L.i} : \mathscr{X} \to [0, 1].$$

Intuitively, the membership defines the degree of possibility that an instance $\mathbf{x} \in \mathscr{X}$ should be propagated down the branch. Without loss of generality, we assume each test to be on a single attribute. Therefore, the membership function is defined over $\texttt{projection}(\mathscr{X}, a_L)$, i.e. the domain of the testing attribute $a_L \in A$.

We further assume that each node N_L is associated with a class C_L and a possibility function P_L.

Definition 4. The possibility function $P_L : \mathscr{X} \to [0, 1]$ is defined by composing the membership functions along the path from the root to node N_L. That is,

$$P_L = \begin{cases} 1 & \text{if } N_L \text{ is the root node,} \\ P_{\hat{L}} \otimes \mu_L & \text{if } N_{\hat{L}} \text{ is the parent of } N_L. \end{cases}$$

The composition operator $\otimes$ is defined in terms of some valid operation for combining two membership functions.

Several composition operators, e.g. fuzzy sum, fuzzy product, and fuzzy max, are supported in our implementation. For example,

$$P_L(\mathbf{x}) = P_{\hat{L}}(\mathbf{x}) + \mu_L(\mathbf{x}),$$

when the fuzzy sum operator is applied.

Fig. 3 shows a sample FCT that classifies instances into two classes C_1 and C_2.

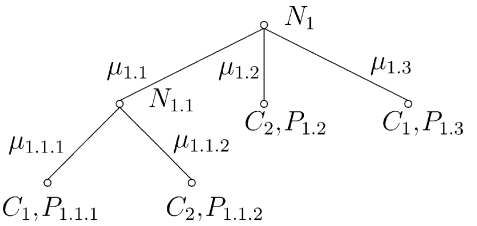

Fig. 3. A sample FCT with $\mathscr{C} = \{C_1, C_2\}$.

Given any instance $\mathbf{x}$ at a terminal node N_L in an FCT, it is classified into class C_L with a possibility $P_L(\mathbf{x})$. As was shown in Fig. 3, multiple terminal nodes may be associated with the same class. It follows that an FCT defines a unique fuzzy classifier

$$\mathbf{F} = \langle \wp_1, \ldots, \wp_n \rangle$$

such that the possibility for an instance belonging to class C_i is the *maximum* over all the possibility values at terminal nodes classified as C_i. That is, for $1 \leqslant i \leqslant n$,

$$\wp_i(\mathbf{x}) = \max\{P_L(\mathbf{x}) \,|\, N_L \text{ is a leaf} \wedge C_L = C_i\}.$$

4. Information-base measure

There are multiple FCTs that implement the same fuzzy classifier. When a classification tree is constructed, choices of attributes to be tested at each node are made. Different attribute selections result in different classification trees. Based on the principle of *simplicity* [7,20,40], it is desirable to create the smallest tree that can correctly classify the most data in the training set. In general, the most discriminating attribute(s) should be chosen first. This section defines the criteria [9] for attribute selection in terms of an information-based measure of FCT.

4.1. Properties of fuzzy entropy functions

Like the decision tree method, one of the important properties of fuzzy classification tree is to identify which attribute is important for classification. According to different position of the attribute in the tree, we can construct many different classification trees. As by Occam's Razor [17,43], the simple one can capture more information than the complex one. It is necessary to determine on choosing which attribute from root to leaf will construct a simple tree.

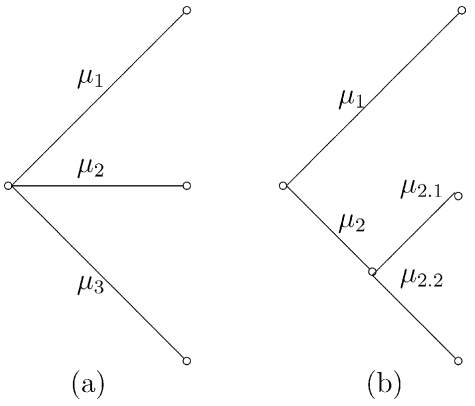

Fig. 4. Decomposition of a choice from three possibilities.

According to the original probabilistic entropy defined by Shannon [40] and fuzzy entropy function defined by De Luca and Termini [20], the information-based measure should satisfy the following criteria. Let the possibility $\wp_i$ for each i define the possibility of an instance, where $\wp_i \in [0,1]$.

[Property 1] Function $H(\wp_1, \wp_2, \ldots, \wp_n)$ should be continuous in $\wp_i$. This property prevents a situation in which a very small change in $\wp_i$ would produce a large (discontinuous) vibration.
[Property 2] Function H must be 0 if and only if all the $\wp_i$ but one are zero. When all but one is possible, there exists no uncertainty in the data.
[Property 3] Function H is the maximum value if and only if the $\wp_i$ are equal because there exists the most uncertainties in the data. That is, no matter what all the $\wp_i$ are, the largest uncertainties happened when all the $\wp_i$ are of the same value.
[Property 4] Function H is a nonnegative valuation on the $\wp_i$.
[Property 5] In order for the purpose that an attribute selection is to reduce the uncertainties in the data, it is necessary that if a choice is broken down into several successive choices, the original H should be no less than the weighted sum of the individual values of H. As illustrated in Fig. 4, in (a) we have three possibilities μ_1, μ_2, and μ_3. In (b), on the right, we first choose between two possibilities, μ_1 and μ_2, and if the second occurs make another choice with possibility $\mu_{2.1}$, $\mu_{2.2}$. The final entropy of (b) should be less than or equal to the entropy of (a). That is

$$H(\mu_1, \mu_2, \mu_3) \geqslant H(\mu_1, \mu_2) + \mu_2 \times H(\mu_{2.1}, \mu_{2.2}).$$

The coefficient μ_2 appears because the second choice is made only with possibility μ_2.

4.2. Fuzzy entropy functions

Suppose we have a set of instances S_L at node N_L. Assume there are n classes associated with the possibilities of occurrences $\wp_1, \wp_2, \ldots, \wp_n$. Concerning about the measure of how much *choice* is involved in the selection of the instance in S_L or of how uncertain we are of the outcome, we choose the entropy function to evaluate that.

Definition 5. The entropy for the set of instances S_L at node N_L is defined by

$$\mathrm{Info}(S_L) = -\sum_{\forall c \in \mathscr{C}} \frac{\mathscr{P}_L^c}{\mathscr{P}_L} \times \log_2 \frac{\mathscr{P}_L^c}{\mathscr{P}_L},$$

where

$$\mathscr{P}_L = \sum_{x \in S_L} P_L(\mathbf{x})$$

is the sum of the possibility value $P_L(\mathbf{x})$ of all instances at node N_L, and

$$\mathscr{P}_L^c = \sum_{\mathbf{x} \in S_L \wedge \mathrm{Class}(\mathbf{x}) = c} P_L(\mathbf{x})$$

is the sum over instances belonging to class c.

The entropy of a set measures the average amount of information needed to identify the class of an instance in the set. It is minimized when the set of instances are homogeneous, and maximized when the set is perfectly balanced among the classes.

A similar measurement can be defined when the set is distributed into b_L subsets, one for each branch based on the test at node N_L. The expected information requirement is the weighted sum over the subsets.

$$\mathrm{Info}_T(S_L) = \sum_{i=1}^{b_L} \frac{\mathscr{P}_{L.i}}{\mathscr{P}_L} \times \mathrm{Info}(S_{L.i}).$$

To asses the "benefits" of a test, we need to consider the increase in entropy. The quality

$$\mathrm{Gain}(\mathrm{Test}_L) = \mathrm{Info}(S_L) - \mathrm{Info}_T(S_L)$$

measures the information gain due to the test Test_L. This gain criterion is used as the basis for attribute selection.

4.3. The requirements of the fuzzy operations

Since the function, $\log_2$ is a continuous function, the fuzzy entropy defined by $\log_2$ is also a continuous function. It is easy to see that Info satisfies Property 1.

If S_L is the set of instances in N_L that has been purely classified into one class, that is all the $\wp_i$ of each instance but one are zero. Let $\wp_i \neq 0$ for some class C_i, then the possibility

$$\mathscr{P}_L = \sum_{x \in S_L} P_L(\mathbf{x}) = \sum_{x \in S_L} \wp_i(\mathbf{x}).$$

The possibilities $\mathscr{P}_L^c$ of the other classes are zero. Because

$$\mathscr{P}_L^c = \sum_{\mathbf{x} \in S_L \wedge \text{Class}(\mathbf{x}) = c} P_L(\mathbf{x}) = 0$$

for $c \neq C_i$. The entropy value of $\text{Info}(S_L)$ will be zero when all the possibilities $\wp_i$ but one are zero.

Property 3 restricts that the entropy value is maximum when all the class possibilities are equal. According to that, it needs that $\sum_c \mathscr{P}_L^c$ should be no bigger than $\mathscr{P}_L$. Otherwise, this property will not be satisfied. Let $|\mathscr{C}|$ be the number of classes and $\mathscr{P}_L^{C_i} = \mathscr{P}_L^{C_j}$ for $i \neq j$. In the FCT algorithm, the sum

$$\begin{aligned}
\text{Info}(S_L) &= -\sum_{\forall c \in \mathscr{C}} \frac{\mathscr{P}_L^c}{\mathscr{P}_L} \times \log_2 \frac{\mathscr{P}_L^c}{\mathscr{P}_L} \\
&\leqslant -\sum_{i=1}^{|\mathscr{C}|} \frac{\mathscr{P}_L}{|\mathscr{C}|\mathscr{P}_L} \log_2 \frac{\mathscr{P}_L}{|\mathscr{C}|\mathscr{P}_L} \\
&= -\sum_{i=1}^{|\mathscr{C}|} \frac{1}{|\mathscr{C}|} \log_2 \frac{1}{|\mathscr{C}|}.
\end{aligned}$$

Operation $\sum$ is defined to be equal to the sum operation in classical (crisp) set.

Since $0 \leqslant \mathscr{P}_L^c \leqslant \mathscr{P}_L$ for all class $c \in \mathscr{C}$, $\log_2 \mathscr{P}_L^c / \mathscr{P}_L \leqslant 0$ and $\text{Info}(S_L) \geqslant 0$. Therefore, it is no doubt that the fourth property is also satisfied.

Why the fifth property is required in the thesis? The purpose of attribute selection in FCTs is toward reducing the uncertainties in the data. After the fuzzy classification tree has been further generating, the total entropy of the child nodes should be no greater than the entropy of their parent nodes. In the other word, the total entropy of child nodes from a node should be less than or equal to the entropy of that node before the tree expanded. That is,

$$\text{Info}(S_L) \geqslant \sum_{i=1}^{b_L} \frac{\mathscr{P}_{L.i}}{\mathscr{P}_L} \times \text{Info}(S_{L.i}).$$

This is a strong constraint that restricts the kinds of fuzzy operations and the membership functions. It also limits the clustering methods to generate the membership function from a node.

Theorem 1. *Let $\otimes$ be the fuzzy t-norm operator. If $\sum_{i=1}^{b_L} \mu_L(\mathbf{x}) \leqslant 1$ for every $\mathbf{x} \in S_L$. Definition 1 satisfies the fifth property of entropy. That is*

$$\text{Info}(S_L) \geqslant \sum_{i=1}^{b_L} \frac{\mathscr{P}_{L.i}}{\mathscr{P}_L} \text{Info}(S_{L.i}).$$

Proof. Let α be the maximal membership value for all membership functions. Let us derive from the right-hand side of the inequality

$$\begin{aligned}
&\sum_{i=1}^{b_L} \frac{\mathscr{P}_{L.i}}{\mathscr{P}_L} \text{Info}(S_{L.i}) \\
&= -\sum_{i=1}^{b_L} \frac{\mathscr{P}_{L.i}}{\mathscr{P}_L} \sum_{c \in \mathscr{C}} \frac{\mathscr{P}_{L.i}^c}{\mathscr{P}_{L.}} \log_2 \frac{\mathscr{P}_{L.i}^c}{\mathscr{P}_{L.i}} \\
&= -\sum_{i=1}^{b_L} \frac{\mathscr{P}_{L.i}}{\mathscr{P}_L} \sum_{c \in \mathscr{C}} \frac{\mathscr{P}_L^c \otimes \mu_{L.i}}{\mathscr{P}_L \otimes \mu_{L.i}} \log_2 \frac{\mathscr{P}_L^c \otimes \mu_{L.i}}{\mathscr{P}_L \otimes \mu_{L.i}} \\
&\leqslant -\sum_{c \in \mathscr{C}} \frac{\mathscr{P}_L^c \otimes \alpha}{\mathscr{P}_L \otimes \alpha} \log_2 \frac{\mathscr{P}_L^c \otimes \alpha}{\mathscr{P}_L \otimes \alpha} \\
&\quad \times \left(\sum_l \mu_l(\mathbf{x}) \leqslant 1 \text{ and } \alpha \geqslant \mu_l(\mathbf{x}),\ \forall l,\ \mathbf{x} \right) \\
&\leqslant -\sum_{c \in \mathscr{C}} \frac{\mathscr{P}_L^c}{\mathscr{P}_L} \log_2 \frac{\mathscr{P}_L^c}{\mathscr{P}_L} \quad \left(\sum_{c \in \mathscr{C}} \mathscr{P}_L^c \leqslant \mathscr{P}_L \right) \\
&= \text{Info}(S_L).
\end{aligned}$$

4.4. Entropy evaluation algorithm

With the above definitions, we can calculate the entropy at the root of an FCT by propagating the entropy values up the tree.

Algorithm *Evaluate_Entropy*
[**Input**] An FCT with root node N_L
[**Output**] The entropy value of $\mathscr{T}_L$

1. $\forall l \in \mathscr{L}$, s.t. N_l is any node in $\mathscr{T}_L$,
   $\text{Info}(S_l) \leftarrow -1$ /* Initialization */
   /* $\text{Info}(S_l)$ is nonnegative, and therefore set a negative value to it first. */
2. $\forall l \in \mathscr{L}$, s.t. N_l is a leaf node,
   $\text{Info}(S_l) \leftarrow -\sum_{c \in \mathscr{C}} \frac{\mathscr{P}_l^c}{\mathscr{P}_l} \times \ln \frac{\mathscr{P}_l^c}{\mathscr{P}_l}$
3. **loop** until $\text{Info}(S_L) \geqslant 0$
   if $\forall i, 1 \leqslant i \leqslant b_l$ $\text{Info}(S_{l.i}) \geqslant 0$ **then**
   $\text{Info}(S_l) \leftarrow \sum_{i=1}^{b_l} \frac{\mathscr{P}_{l.i}}{\mathscr{P}_l} \times \text{Info}(S_{l.i})$
   end
4. **return** $\text{Info}(S_L)$.

5. Construction

This section presents the learning algorithm for constructing a fuzzy classification tree from a set of training instances containing real-valued attributes. Previous approaches to this problem usually fuzzify the data before they are used to construct a decision tree [47]. The linguistic variables have to be defined ahead of time based on existing domain knowledge.

The main algorithm for FCT construction takes an input a set S_0 of instances, and starts by creating a root node N_1, adding its label to $\mathscr{L}$, and initializing S_1 to be S_0.

Algorithm *Build_FCT*
[**Input**] A set of training instances S_0
[**Output**] An FCT

1. $L \leftarrow 1$
   /* Initialize L to be 1 which is the label at the root node. */
2. $\mathscr{L} \leftarrow \{1\}$
   /* Let $\mathscr{L}$ be the set of labels represented the nodes that have not been expanded. */
3. $S_1 \leftarrow S_0$
   /* S_1 at the root node is set to be the original set S_0. */
4. **loop** until $\mathscr{L} = \phi$
5. $L \leftarrow \text{random}(\mathscr{L})$
   /* Random select one of the label from $\mathscr{L}$. */
6. $\mathscr{L} \leftarrow \mathscr{L} \backslash \{L\}$
7. $\forall a_i,\ \tau_i \leftarrow \text{Spawn\_New\_Tree}(N_L, a_i)$
8. Find τ_k s.t. $\text{Info}(\tau_k) = \max_j \text{Info}(\tau_j)$
9. $\text{Gain} \leftarrow \text{Info}(\mathscr{T}_L) - \text{Info}(\tau_k)$
10. **if** $\text{Gain} > \varepsilon$ **then**
    $\mathscr{L} \leftarrow \mathscr{L} \cup \text{leaf}(\tau_k)$
    Assign subsets of S_L into $S_{L.1}, \ldots, S_{L.k}$.

The procedure $\text{Spawn\_New\_Tree}(N_L, a_i)$ expands the tree from node N_L according to some attribute a_i.

6. Clustering

The membership function is the kernel for fuzzy classifications. To determine the membership function from a data set, the method of clustering is used. Clustering is a well-used method in pattern recognition. It plays a key role in searching for structures in data. There may be different kinds of models simultaneously occurring in the data, that is called *multi-model* [5]. Data could be clustered into differential groups in accordance to their distribution models. The models construct the membership function of the data.

Algorithm *Spawn_New_Tree*
[**Input**] An unexpanded node N
An attribute a
[**Output**] An expanded tree rooted at node N
$\forall i,\ 1 \leqslant i \leqslant n$ do the following:

1. *Project* instances at node N of class C_i onto attribute a.
2. *Smooth* the resulting histogram using k-median method.
3. *Partition* the smoothed histogram into clusters.
4. *Create* a new branch from N_L for each cluster.
5. *Define* the membership function for each branch.

6.1. Clustering on numerical attributes

Clustering is the important operation to deal with real-valued attributes to derive the membership function from them, as well as symbolic attributes. Deriving the memberships of symbolic attributes will be introduced later. For real-valued attributes,

it can be used to partition the domain of each one of them into several clusters according to its distribution. Given a finite set of data, X, of a real-valued attribute, clustering in X is to find several cluster centers that can properly characterize relevant categories of X. In classical approaches, these categories are required to form a partition of X. Each instance in X is uniquely assigned to a categories of the partition. However, this requirement is too strong in many practical applications, such as medical diagnosis, financial management, robot control, etc. Because of the uncertainties, that make partition boundaries not so clear. There exists an uncertain overlapped region at each partition boundary. It is thus desirable to shift it with a weaker requirement to describe this overlapped situation.

Fuzzy c-means clustering method, which satisfies the weaker requirement, is used to make a properly vague partition. The membership value of each datum defines how possible this datum is associated with a category. The membership gives a meaningful explanation on this vagueness. Therefore, to deal with the unavoidable observation and measurement uncertainties, fuzzy clustering is a very suitable choice applied to real world applications.

6.1.1. Fuzzy c-means clustering method

No universally optimal clustering criteria existed can efficiently group a data set into clusters. Bezdek [1] has proposed the fuzzy c-means clustering method to solve this optimization problem.

Given a data set $X=\{x_1,x_2,\ldots,x_n\}$, a fuzzy c-partition of X is a family of fuzzy subsets of X, denoted by $\mathbf{P}=\{\mu_1,\ldots,\mu_c\}$, where $c\in\aleph$ and

$$\sum_{k=1}^{c}\mu_k(x_i)=1$$

for all $i\in\{1,2,\ldots,n\}$ and

$$0<\sum_{i=1}^{n}\mu_k(x_i)<n$$

for all $k\in\{1,2,\ldots,c\}$. The membership function $\mu_i\in\mathbf{P}$, $1\leqslant i\leqslant c$, denotes the function for evaluating the degree of uncertainties of X in class C_i. For instance, given $X=\{x_1,x_2,x_3\}$ and

$$\mu_1=.6/x_1+0/x_2+.2/x_3,$$

$$\mu_2=.4/x_1+1/x_2+.8/x_3,$$

$\{\mu_1,\mu_2\}$ is the fuzzy 2-partition of X.

The fuzzy c-means clustering method requires a criterion that the association of the data is strong within a cluster and weak between clusters. Let $v_1,v_2,\ldots,v_c$ be c cluster centers of $\mathbf{P}$. Each cluster center associated with the partition is calculated by the following formula.

$$v_k=\frac{\sum_{i=1}^{n}[\mu_k(x_i)]^m x_i}{\sum_{i=1}^{n}[\mu_k(x_i)]^m},$$

where $1\leqslant k\leqslant c$, $m>1$ is a real number that governs the influence of membership grades. The weight of a datum x_i is the mth power of the membership grade $\mu_k(x_i)$. When $m\to 1$, the fuzzy c-means converges to a *generalized* classical c means. When $m\to\infty$, all cluster centers tend towards the centroid of the data set X. That is, the partition becomes fuzzier with increasing m.

Definition 6 (Bezdek). The criterion of a fuzzy c-partition is defined in terms of the cluster centers by the formula

$$\sum_{j=1}^{n}\sum_{k=1}^{c}[\mu_k(x_j)]^m\,\|x_j-v_i\|^2,$$

where $\|\cdot\|$ is the inner product-induced norm in $\Re$ and $\|x_j-v_i\|^2$ represents the distance between x_j and v_i.

The goal of the fuzzy c-means clustering method is to find a fuzzy partition S that minimizes the criterion.

7. Empirical results

We have tested our algorithm on five data sets from the UCI repository (ftp://ftp.ics.uci.edu/machine-learning-databases).

Glass: This is sample data set including 214 instances in determining whether the glass was a type of "float" glass or not for criminological investigation. Seven classes with nine numerical attributes with missing data.

Table 2
Average accuracy between C4.5 and FCT over the glass, monks, and ionosphere data sets

	Glass	Monk1	Monk2	Monk3	Ionosphere
C4.5	$94.5 \pm 3.3\%$	$77.1 \pm 3.3\%$	$65.3 \pm 6.7\%$	$92.6 \pm 2.9\%$	$95.5 \pm 2.5\%$
FDT (Hsu et al.)	$95.2 \pm 3.4\%$	$78.9 \pm 3.4\%$	$68.1 \pm 6.8\%$	$93.3 \pm 2.2\%$	$95.5 \pm 2.3\%$
FDT (Yuan and Shaw)	$95.3 \pm 3.1\%$	$80.6 \pm 3.4\%$	$68.7 \pm 7.0\%$	$92.7 \pm 1.9\%$	$95.1 \pm 2.0\%$
FDT (Janickow)	$95.6 \pm 3.3\%$	$85 \pm 2.1\%$	$71.7 \pm 6.3\%$	$93 \pm 1.9\%$	$95.4 \pm 2.4\%$
FDT (Suárez and Lutsko)	$93.4 \pm 4.0\%$	$84.6 \pm 2.7\%$	$70.8 \pm 5.3\%$	$94.5 \pm 1.6\%$	$94.7 \pm 2.2\%$
FCT	$96.2 \pm 2.8\%$	$86.2 \pm 2.9\%$	$73.4 \pm 5.8\%$	$93.2 \pm 1.7\%$	$95.3 \pm 1.8\%$

Monks' problem: The three Monks' problems are a collection of three binary classification problems over a six-attribute discrete domain. The classes is either 0 or 1. Six categorical value attributes, no missing value. There are noisy data in *monk1* and *monk2*.

Ionosphere: This data set is a binary classification task. The radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kW. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. There are 351 instances with all 34 numerical attributes, no missing value.

The result of comparing the accuracy of FCT with C4.5 and three kinds of fuzzy decision trees which are proposed by Hsu et al. [15], Yuan and Shaw [47], Janickow [18] and Suárez and Lutsko [41] on these problems is shown in Table 2. Note the tree that we use to compare is without pruning.

All those data sets were tested according to the F-test under the confident level of 95% by 5-folded cross validation.

The clustering method determines the performance of FCT. The clustering method used for classifications is fuzzy c-partition algorithm. It full satisfies the criteria of possibilistic entropy. As the result on the *ionosphere* problem, the accuracy of FCT is lower than the accuracy of C4.5 and the fuzzy decision trees proposed by Hsu et al. Since the values of all the attributes are distributed in $[-1, 1]$, the size of the clusters has an effect on the accuracy of FCT. To improve the clustering method is one of our further objectives. The pre-fuzzification methods proposed by Yuan and Shaw are worse than FCTs. Because Yuan and Shaws' algorithms partition the numerical data first and then use the linguistic data to construct the decision tree, therefore, it is unavoidable that the performance of the clustering methods affects the accuracy of these algorithms much more than FCTs. The lack of dynamic clustering adjustment as FCTs makes the Yuan and Shaws' algorithms less accurate.

8. Discussion

Classification by decision trees has been successfully applied to problems in artificial intelligence, pattern recognition and statistics. However, as Quinlan [31] pointed out "the results of decision trees are categorical and so do not convey potential uncertainties in classification". Missing or imprecise information may prevent a case from being classified at all. In the presence of uncertainties, it is often desirable to have an estimate of the degree that an instance is in each class, e.g. medical diagnosis.

Instead of classifying a case as belonging to exactly one class, and ruling out the other possibilities, one can estimate the relative probabilities of its being in each class. Casey and Nagy [6] designed a decision tree classifier using probabilistic model for optical character recognition process. Breiman et al. [3] introduced the class probability estimate. Quinlan [31,34] proposed probabilistic decision trees to deal with uncertainties in data. Schuermann and Doster [39] also proposed using probabilistic model to estimate the probability of each class. In addition, to deal with search bias introduced in attribute selection and hypotheses-space bias due to noisy data [4], Buntine [5] suggested averaging over multiple class probability trees.

Probabilistic approaches still assume that there is only one decision node in the tree to which a case can be classified. A test instance falls down a single branch to arrive at a leaf where a probability is associated with each class. Such classifications ignore the information at the other nodes. However, several methods, including Buntine's classification trees [5], Rymon's *set enumeration* tree [36] have been proposed to address this issue. However, the approaches are inefficient in both time and space.

In a fuzzy classification tree, an instance has a membership value at each leaf node. We can calculate the degree of possibility that the instance belongs to any of the classes. Using information-based measures, there is no need to generate multiple classification trees. Therefore, it requires less time and space than decision forests.

9. Conclusion

This paper has presented an algorithm that integrates the fuzzy classifiers with decision trees. The algorithm attempts to expand the FCT while minimizing its entropy at each step.

Unclearly partitioned boundaries between classes strongly confuse the conclusion obtained from C4.5. However, overly generalized predictions are the serious problem [42,47]. Opposite to C4.5, fuzzy classification trees give much better predictive conclusion. Fuzzy classification predicts the degree of *possibility* for every class instead of determining a single class for any given instance. According to these possibilities, a proper conclusion can be made for each instance.

We have compared FCT with C4.5 and four kinds of fuzzy decision trees with the empirical results of five data sets in the above section. From the noise-free data (Golf) to the data with a great amount of noise (Monk2), the accuracy rate of FCT is better than those. C4.5 classifies an instance into exactly one class. The instances with attribute values around class boundaries are forced to be classified into a single class, which may result in wrong predictions, especially in the noisy domains. Instead of making a rigid classification, it is sometimes necessary to identify more than one possible classifications for a given instance. Although the four fuzzy decision trees provide multiple classification for an instance, they do not consider the distribution of the data that can make an improper classifications. FCTs can properly solve this problem. Through fuzzy clustering, the data could reveal the "context" structure for each attribute, as pointed out by Pedrycz and Sosnowski [26]. Based on the parent node's testing attribute, the cluster structure could give the globally optimal decision for each node.

FCTs allow multiple predictions to be made, each of which is associated with a degree of possibility. In application domains that involve a large amount of data with uncertainty, such as medicine or business, fuzzy classification trees can serve as a useful tool for generating fuzzy rules or discovery knowledge in database.

References

[1] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[2] X. Boyen, L. Wehenkel, Automatic induction of fuzzy decision trees and its application to power system security assessment, Fuzzy Sets and Systems 102 (1999) 3–19.
[3] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and Regression Trees, Chapman & Hall, London, 1984.
[4] W. Buntine, Myths and legends in learning classification rules, Proc. 8th National Conf. on Artificial Intelligence, Boston, MA, 1990, pp. 736–742.
[5] W. Buntine, Learning classification trees, Statist. Comp. 2 (1992) 63–73.
[6] R.G. Casey, G. Nagy, Decision tree design using a probabilistic model, IEEE Trans. Information Theory 30 (1) (1984) 93–99.
[7] R.L.P. Chang, T. Pavlidis, Fuzzy decision tree algorithms, IEEE Trans. Syst. Man Cybern. 7 (1) (1977) 28–35.
[8] Z. Chi, H. Yan, ID3-derived fuzzy rules and optimized defuzzification for handwritten numeral recognition, IEEE Trans. Fuzzy Syst. 4 (1) (1996) 24–31.
[9] I. Chiang, J. Hsu, Integration of fuzzy classifiers with decision trees, Proc. Asian Fuzzy Syst. Symp., Kenting, Taiwan, 1996, pp. 65–78.
[10] K.J. Cios, L.M. Sztandera, Continuous ID3 algorithm with fuzzy entropy measures, Proc. Int. Conf. on Fuzzy Systems, San Diego, CA, 1992, pp. 469–476.
[11] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, Proc. 12th Int. Conf. on Machine Learning, San Mateo, CA, 1995, pp. 194–202.
[12] U.M. Fayyad, K.B. Irani, On the handling of continuous-valued attributes in decision tree generation, Machine Learning 8 (1992) 87–102.

[13] D. Heath, S. Kasif, S. Salzberg, Learning oblique decision trees, Proc. 13th Int. Joint Conf. on Artificial Intelligence, Chambery, France, 1993, pp. 1002–1007.
[14] J.Y. Hsu, I. Chiang, Fuzzy classification trees, Proc. 9th Int. Symp. on Artificial Intelligence, Cancun, Mexico, 1996, pp. 431–438.
[15] S. Hsu, J.Y. Hsu, I. Chiang, Automatic generation of fuzzy control rules by machine learning methods, Proc. Int. Conf. on Robotics and Automation, Nagoya, Japan, 1995, pp. 287–292.
[16] E.B. Hunt, J. Marin, P.J. Stone, Experiments in Induction, Academic Press, Orlando, FL, 1996.
[17] A. Hyman, J.J. Walsh, Philosophy in the Middle Ages, 2nd ed., Hackett Publishing Co., Indianapolis, 1973.
[18] C.Z. Janickow, Fuzzy decision trees: issues and methods, IEEE Trans. Syst. Man Cybern. B: Cybern. 28 (1) (1998) 1–14.
[19] I. Kononenko, I. Bratko, E. Roskar, Experiments in automatic learning of medical diagnostic rules, Technical Report, Jozef Stefan Institute, Ljubljana, Yugoslavia, 1984.
[20] A. De Luca, S. Termini, A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory, Inf. Control 20 (1976) 301–312.
[21] P.E. Maher, D. St. Clair, Uncertain reasoning in an ID3 machine learning framework, Proc. 2nd IEEE Int. Conf. on Fuzzy Systems, San Francisco, CA, 1993, pp. 7–12.
[22] S.K. Murthy, On growing better decision trees from data, Ph.D. Dissertation, The Johns Hopkins University, Baltimore, Maryland, 1995.
[23] S.K. Murthy, S. Kasif, S. Salzberg, A system for induction of oblique decision trees, J. Artif. Intell. Res. 2 (1994) 1–32.
[24] S.K. Murthy, S. Kasif, S. Salzberg, R. Beigel, OC1: randomized induction of oblique decision trees, Proc. 11th National Conf. on Artificial Intelligence, Washington, DC, 1993, pp. 322–327.
[25] Z. Pawlak, Rough Sets, Kluwer Academic, Dordrecht, 1991.
[26] W. Pedrycz, Z.A. Sosnowski, The design of decision trees in framework of granular data and their application to software quality models, Fuzzy Sets and Systems 123 (2001) 271–290.
[27] J.R. Quinlan, Discovery rules by induction from large collections of examples, in: D. Michie (Ed.), Expert Systems in the Micro Electronic Age, Edinburgh University Press, Edinburgh, UK, 1979.
[28] J.R. Quinlan, Learning efficient classification procedures and their application to chess endgames, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning: An Artificial Intelligence Approach, Tioga, Palo Alto, CA, 1983.
[29] J.R. Quinlan, The effect of noise on concept learning, in: R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning, Morgan Kaufman, Los Altos, CA, 1985.
[30] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[31] J.R. Quinlan, Probabilistic decision trees, in: P. Langley (Ed.), Proc. 4th Int. Workshop on Machine Learning, Los Altos, CA, 1987.
[32] J.R. Quinlan, Simplifying decision trees, Int. J. Man-Machine Studies 27 (1987) 221–234.
[33] J.R. Quinlan, Decision trees and decision making, IEEE Trans. Syst. Man Cybern. 20 (1990) 339–346.
[34] J.R. Quinlan, Learning logical definitions from relations, Machine Learning 5 (1990) 239–266.
[35] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993.
[36] R. Rymon, An SE-tree based characterization of the induction problem, Proc. 10th Int. Conf. on Machine Learning, Amherst, MA, 1993, pp. 268–275.
[37] S.R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, IEEE Trans. Syst. Man Cybern. 21 (3) (1991) 660–674.
[38] J.C. Schlimmer, R.H. Granger Jr., Incremental learning from noisy data, Machine Learning 1 (1986) 317–354.
[39] J. Schuermann, W. Doster, A decision theoretic approach to hierarchical classifier design, Pattern Recognition 17 (3) (1984) 359–369.
[40] C.E. Shannon, A mathematical theory of communication, The Bell Syst. Technical J. 27 (1948) 379–423, 623–656.
[41] A. Suárez, J.F. Lutsko, Globally optimal fuzzy decision trees for classification and regression, IEEE Trans. Pattern Anal. Machine Intell. 21 (12) (1999) 1297–1311.
[42] M. Sugeno, G.T. Kang, Structure identification of fuzzy model, Fuzzy Sets and Systems 28 (1988) 15–33.
[43] W.M. Thorburn, The myth of occam's razor, Mind 27 (1918) 345–353.
[44] Q.R. Wang, C.Y. Suen, Large tree classifier with heuristic search and global training, IEEE Trans. Pattern Anal. Machine Intell. 9 (1) (1987) 91–102.
[45] R. Weber, Automatic knowledge acquisition for fuzzy control application, Proc. Int. Symp. Fuzzy Systems, Iizuka, Japan, 1992, pp. 9–12.
[46] R. Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1992, pp. 265–268.
[47] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.