International Journal of Fuzzy Logic and Intelligent Systems, vol. 3, no. 1, June 2003 pp. 44-51

Fuzzy Classification Rule Learning by Decision Tree Induction

Keon Myung Lee[†] , **Hak-Joon Kim**[‡]

[†] **School of Electric and Computer Engineering,**
Research Institute for Computer and Information Communication, Chungbuk National University
and Advanced Information Technology Research Center(AITrc), Korea
[‡] **Div. of Electrical, Electronic, Information, Communication Engineering,**
Howon Univeristy, Korea

Abstract

Knowledge acquisition is a bottleneck in knowledge-based system implementation. Decision tree induction is a useful machine learning approach for extracting classification knowledge from a set of training examples. Many real-world data contain fuzziness due to observation error, uncertainty, subjective judgement, and so on. To cope with this problem of real-world data, there have been some works on fuzzy classification rule learning. This paper makes a survey for the kinds of fuzzy classification rules. In addition, it presents a fuzzy classification rule learning method based on decision tree induction, and shows some experiment results for the method.

Key words : *fuzzy classification rules, inductive learning, decision tree*

1. Introduction

Classification is an important operation in pattern recognition, data analysis, decision making, and many other applications. A lot of human knowledge can be represented in classification rules. Knowledge acquisition is a crucial step for knowledge-based system implementation and it usually becomes a bottleneck. To cope with this kind of bottleneck, there have been many works on automatic knowledge acquisition.[1] Decision tree induction is one of such works, which extracts classification knowledge by constructing decision trees from a set of examples.

Decision trees denote tree structures that represent classification rules. In decision trees, non-terminal nodes have the information about feature to be compared for classification and terminal nodes are labeled with the class to which the data satisfying all conditions on the path from the root node to the terminal node. Decision trees can be easily transformed into IF-THEN classification rules. There have been many works on decision tree induction methods to create small and accurate decision trees.[1]

Due to observation error, uncertainty, subjective judgement, and so on, many real world data inherently contain some noises and fuzziness in themselves. In the decision tree literature, various researches have been done on fuzzy decision trees to make decision trees less sensitive to such noises and fuzziness. Fuzzy decision trees denote the decision trees which incorporate fuzzy techniques in their representation and construction. Fuzzy decision trees represent fuzzy classification rules. So far various kinds of fuzzy decision tree induction methods have been developed.[2,3,4,8,10,12]

To begin with, this paper presents what kinds of fuzzy classification rules can be composed when fuzzy techniques are incorporated into and what characteristics they have. Then it proposes a fuzzy classification rule learning method based on a fuzzy decision tree induction method. The proposed classification rule learning method produces fuzzy classification rules from a set of crisp data with numeric and nominal attributes.

This paper is organized as follows: Section 2 presents the types of fuzzy rule-based classification systems. Section 3 introduces a decision tree induction method to extract fuzzy classification rules from a set of training data. Section 4 shows some experiment results for the proposed method and finally Section 5 draws conclusions.

2. Fuzzy Rule-based Classification Systems

Classification systems can be implemented in various ways such as rule-based systems, artificial neural networks, statistical model-based systems, instance-based methods, and so on. The rule-based systems are the most popular approach which represents classification knowledge in IF-THEN rules. The artificial neural network models such as multi-layer perceptron, ART, Kohonen's feature map have been successfully applied to the domains in which it is difficult to extract classification knowledge explicitly. The statistical models like native Bayesian classifier, Bayes optimal classifier are a classical approach to use the statistic estimates based on the given training data set. The instance-based methods like the *k*-nearest neighbor algorithm, the case-based reasoning

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center (AITrc)

method, and so on extract typical examples for each class and use them to classify new data. Among the above various implementation approaches, the rule-based systems are most widely used since they can be easily understood and can easily represent human classification knowledge.

In rule-based classification systems, their knowledge base consists of a set of IF-THEN classification rules as follows:

R_1 : IF $condition_1$ THEN $class_1$
R_2 : IF $condition_2$ THEN $class_2$
$\vdots$
R_n : IF $condition_n$ THEN $class_n$

Fig.1 A knowledge base of rule-based classification system

In the above classification rules R_i, the antecedent $condition_i$ specifies the properties that a data possesses, and the consequent $class_i$ indicates the class to which data satisfying the antecedent conditions belong. As an effort to deal with fuzziness inherent in data and to improve classification accuracy, several works have been done on incorporating fuzzy techniques to rule-based classification systems.[8,9,10,11] The incorporation of fuzzy techniques enables to express more flexibly classification rules and to give the chance to infer more accurate results.

Depending on the types of antecedent predicates and consequent labels, various types of classification rules might be developed. They can be categorized into 5 types of classification rules as follows:

Type 1 classification rules:
IF $cp(x)$ THEN $class = c_i.$ $[cf_i]$
Type 2 classification rules:
IF $cp(x)$ THEN $class = fset_i.$
Type 3 classification rules:
IF $fp(x) \wedge [cp(x)]$ THEN $class = c_i.$ $[cf_i]$
Type 4 classification rules:
IF $fp(x) \wedge [cp(x)]$ THEN $class = flt_i.$
Type 5 classification rules:
IF $fp(x) \wedge [cp(x)]$ THEN $class = fset_i.$

In the above, $cp(x)$ denotes a (crisp) predicate to map input x to the truth value set $\{0,1\}$, $fp(x)$ denotes a fuzzy predicates to map input x to the range $[0,1]$, and cf_i is the certainty factor. The singleton value '$class = c_i$' is the most widely used representation for consequent. '$class = fset$ is a fuzzy set of classes and '$class = flt_i$' is fuzzy linguistic terms on a class domain. In the rules, [] means that the corresponding terms are optional.

Types 1 and 2 classification rules are called crisp classification rules. Types 3, 4, and 5 classification rules are called fuzzy classification rules. Compared to crisp classification rules, fuzzy classification rules are advantageous in that they may be robust to noise of data values and provide more meaningful classification information since they can assign multiple classes with different membership degrees to data located near class boundaries in the feature space.

Classification rules can be acquired in various ways. The most straightforward approach is that knowledge engineers manually construct classification rules from human expertise. This requires lots of knowledge engineering know-how to generate good classification rules. In addition, it is difficult to construct complete classification rules to cover the entire feature space with this approach. On the other hand, many works on classification rule learning have been conducted to extract classification rules from a collection of training data. As the representative ones, there are decision tree induction, neural network learning, genetic algorithm-based learning, and so on. Some rule learning methods use crisp data sets, and others use fuzzy data sets of which data contains fuzzy values. Most classification rule learning methods focus on crisp classification rule learning from crisp training data set. Fuzzy classification rule learning has not yet been studied so much as crisp classification rule learning.

Decision trees are well-known representation structures which can easily represent classification rules. Many decision tree induction methods have been developed that extract decision trees from a set of training examples. In addition, there are several fuzzy decision tree induction methods that construct fuzzy decision trees which correspond to fuzzy IF-THEN classification rules. The following section introduces a fuzzy classification rule learning method based on fuzzy decision tree induction.

3. Fuzzy Classification Rule Learning

This section presents a fuzzy decision tree induction method to produce Type 3 fuzzy IF-THEN classification rules of which antecedent may have fuzzy numeric predicates and/or crisp nominal predicates, and of which consequent is a fuzzy set of classes. It is assumed that the training data set consists of crisp data whose class label comes together a confidence degree ranged on [0,1].

A data D_i of an object or an event is assumed to be represented with a set of attributes describing the object like this: Here A^i_j denotes the i-th attribute for the object.

$$D_i = (A^i_1, A^i_2, \cdots, A^i_n)$$

Some attributes have discrete nominal domain, and others have continuous numeric domain. It is assumed that both discrete nominal domain attributes and continuous numeric domain attributes can be characterized by crisp values.

3.1 A Fuzzy Decision Tree Induction Method

3.1.1 Fuzzy Decision Trees

Decision trees classify data by sorting them down the tree from the root to leaf nodes. As the typical kinds of decision tree induction algorithms, there are ID3 and CART[8][13]. ID3 is designed to deal with symbolic nominal domain data, and CART is designed to deal with continuous numeric domain data. Several alternations of them have been reported in the literature[1]. Fuzzy decision trees are one of them.

In classical decision trees, nodes make data follow down only one branch since data satisfies a branch condition, and the data finally arrives at only a leaf node. On the other hand, fuzzy decision trees allow data to follow down simultaneously multiple branches of a node with different satisfaction degrees ranged on [0,1]. To implement these characteristics, fuzzy decision trees usually use fuzzy linguistic terms to specify branch conditions of nodes.

This section introduces a new fuzzy decision tree induction method with the following properties: First, its training data set is made up of crisp data of which nominal attributes and numeric attributes are represented by crisp values, and of which class is represented by a nominal value with confidence degree ranged on the interval [0,1]. Second, the decision tree induction method generates fuzzy membership functions for continuous numeric attributes on the fly. Thus, it is unnecessary to specify fuzzy membership function before applying the tree induction method as in [8]. Third, the tree induction method can control the branching factor of decision tree to be constructed at will. Due to this, nodes corresponding to nominal attribute may have sets of nominal values on their branches.

Figure 1 exemplifies a fuzzy decision tree that the fuzzy decision tree induction method to be presented might generate. The root node corresponds to the nominal attribute *Job* and its branches are labeled with crisp sets of attribute values : {A,D}, {B}, and {A,C}. Numeric attribute nodes like *Salary* have fuzzy linguistic terms like *low* and *high* of which membership functions are constructed during the tree-building process. As we can see in the figure, leaf nodes of fuzzy decision trees do not always contain only one class label and in addition class membership degree of nodes are not always 1. Leaf nodes may have several classes with different membership degrees. A data fed into the root node may reach multiple leaf nodes. For the final class assignment of data, the classification information of all leaf nodes needs to be aggregated.

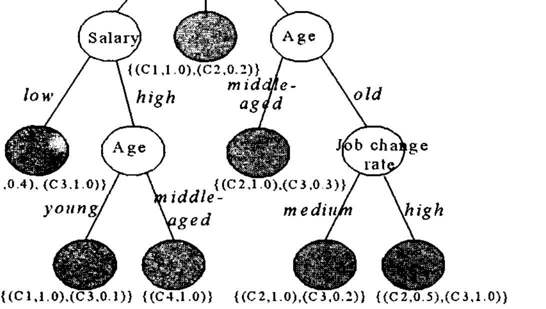


Fig.2 A fuzzy decision tree

Fuzzy decision tree induction systems have two major components: a procedure for fuzzy decision tree building and an inference procedure for decision making(i.e., class assignment of new data). The proposed fuzzy decision tree building procedure takes a

similar approach to the well-known ID3 algorithm[1] that constructs decision tree by recursive partitioning of data set according to the values of selected attribute. It is required to develop the following things to apply an ID3-like procedure to fuzzy decision tree construction: attribute value space partitioning methods, branching attribute selection method, branching test method to determine with what degree data follows down branches of a node, and leaf node labeling methods to determine classes for which leaf nodes stand. This section presents the above-mentioned things for the proposed fuzzy decision tree building procedure, and an inference procedure for class assignment of new data.

3.1.2 Attribute Value Space Partitioning

To avoid the explosive growth of fuzzy decision tree(i.e., the number of classification rules), the branching factor of the constructed decision tree should be controllable. It means that the fuzzy decision tree building procedure should be able to partition attribute value spaces into any desired number of partitions. The proposed fuzzy decision tree building procedure partitions nominal attribute value spaces into sets of nominal attribute values, and continuous numeric attribute value spaces into trapezoidal fuzzy numbers[14].

For nominal attribute value space partitioning, we introduce a new concept called classwise element set which is defined as follows:

$$CWS_i^k(A) \;=\; \{D_j.A | D_j.class = i, \mu_{N_k}(D_j) > 0\}$$

Here $\{ CWS_i^k(A)\}$ denotes the classwise element set for class i at node N_k with respect to a nominal attribute A. $D_{j.A}$ is the value of attribute A of data D_j, $D_j.class$ is the class to which data D_j belongs, and $\mu_{N_k}(D_j)$ is the membership degree of data D_j to N_k. Suppose that the set of pairs $(D_j.A, D_j.class)$ of the attribute value $D_j.A$ and class $D_j.class$ of data arriving at N_k is like this: $\{(a,1),(b,1),(b,2),(c,1),(c,2),(c,3),(d,2),(d,4),(e,3),(e,4)\}$. For the node N_k, the classwise element sets are as follows:

$$CWS_1^k(A) = \{a, b, c\},\;\; CWS_2^k(A) = \{b, c, d\}$$

The classwise element set $CWS_i^k(A)$ implies that data whose attribute A has a value in $CW_i^k(A)$ at node N_k probably belongs to class i. In this perspective, we use classwise element sets $CWS_i^k(A)$ as bases for nominal attribute value space partitioning.

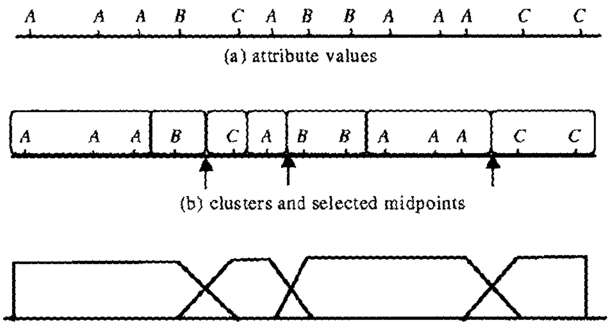
When the number of classes c is not less than the branching factor b of fuzzy decision tree, the partitioning method considers possible partitioning combinations based on $CWS_i^k(A)$ to divide c classwise element sets into b groups.

Among such partitioning combinations, it chooses the combination with the best performance. For example, in the above, the partitioning combination $CWS_1^k(A), CWS_2^k(A)$ and $\{ CWS_3^k(A)\ CWS_4^k(A)\}$ forms two partitioning sets $\{ a, b, c, d\}$ and $\{ c, d, e\}$ for the value space of attribute A. When the branching factor is greater than the number of classes, some special treatment is needed. The following is the procedure for nominal attribute value space partitioning.

```
procedure : nominal attribute space partitioning
  input : the set of attribute values of which attribute space
          to be partitioned, and branching factor $bf$
  output : partitioning combination consisting of sets of
           attribute values, and information gain
begin
  Find the classwise element sets $CWS$ for the data set of
     the current node.
  if (the size of classes $\geq$ the branching factor $bf$) then
      Generate all possible partitioning combinations to make
        $CWS$ into $bf$ groups.
      Calculate the information gain for each partitioning
        combination.
      Return the partitioning combination with the largest
        information gain.
  else
      Set the initial partitioning combination with $CWS$.
      Calculate the impurity measure for each partition.
      do
          Find the most impure partition $MC$.
          Generate all possible partitioning combinations to
             divide $MC$ into two partitions.
          Remove all combinations containing an already
             existing partition.
          Calculate the impurity measure for each partitioning
             combination.
          Select the best combination of which partitions are
             different from existing partitions.
          Replace $MC$ with the two partitions of the best
             partitioning combination.
      until to have $b$ partitions
      Return the final partitioning combination with its
        information gain.
  endif
end.
```

For numeric attribute value space partitioning, we use trapezoidal fuzzy numbers to form fuzzy partitions. We may generate infinitely many number of fuzzy partitioning combinations for a numeric attribute value space. In the proposed fuzzy decision tree induction method, we use the following strategy to construct fuzzy linguistic terms (i.e., trapezoidal fuzzy numbers) for numeric attribute value space.

It sorts the attribute values in the ascending order. In the sorted sequence, it is reasonable to locate fuzzy partitioning boundaries at the midpoints between adjacent attribute values belonging to different classes each other. The number of possible combination to choose such midpoints as many as branching factor bf (in fact, bf - 1) is huge. When there are many attribute values, it is intractable to consider all possible such combinations. In this reason, it takes a stochastic approach like this: It merges adjacent attribute values belonging to the same class into a cluster, and calculates the interval length that each cluster extends. It assigns a selection probability to each midpoint of clusters, which is proportional to the interval length sum of clusters corresponding to the midpoint. Then it generates the prespecified number of fuzzy partitions according to the selection probabilities. The number of generated fuzzy paritions can be controlled at will in the application fields. It chooses the best one among the generated fuzzy partitions.



(a) attribute values

(b) clusters and selected midpoints

(c) fuzzy partition

Fig. 3 Numeric attribute value space partitioning

Figure 3 shows an example to make fuzzy partition for the given attribute values. Figure 3-(a) shows attribute values and Figure 3-(b) shows the clusters where upward arrows indicate the selected midpoints. Figure 3-(c)presents the fuzzy partition corresponding to the selected midpoints.

```
procedure : numeric attribute value space partitioning
  input : the set of attribute values of which attribute
          space to be partitioned, and branching factor
  output : a fuzzy partition and its information gain
begin
  Sort the attribute values in the ascending order. Let the
     sorted attribute value sequence be $SQ$ =($p_1$, $p_2$,
     ..., $p_n$).
  Make the adjacent attribute value points belonging to the
     same class into a cluster. A cluster $Cl_k$ is a set
     such that $Cl_k = \{ p_i, p_{i+1}, ..., p_j\}$ belongs to the
     same class and their class is different from $p_{i-1}$'s
     class and $p_{j+1}$'s class.
  Calculate the diameter of each cluster. The diameter $L_k$
     of a cluster $Cl_k = \{ p_i, p_{i+1}, ..., p_j\}$ denotes the
     interval length of $[(p_{i-1}+p_i)/2, (p_j+p_{j+1})/2]$.
  Set the probability $\Pr_i$ for the boundary point of
     clusters $Cl_i$ and $Cl_{i+1}$ to be selected as follows:
     $AC_i = (L_i + L_{i+1})/2, \ \Pr_i = AC_i / \sum_j AC_j$.
```

```
    Generate the pre-specified number $t$ of sets $PS_k$ that consists of $b-1$ boundary points which are randomly selected according to the selection probability $\Pr_i$.
    Construct a fuzzy partition $FP_k$= {Trap($a_k$, $b_k$, $c_k$, $d_k$) | $k$ = 1, ..., $b$} for each set $PS_k = \{p_1^k, p_2^k, \ldots, p_{b-1}^k\}$ as follows: $a_k = c_{k-1}$. $b_k = d_{k-1}$. $c_k$ is the rightmost point belonging to the left cluster corresponding to boundary point $p_1^k$. $d_k$ is the leftmost point belonging to the right cluster corresponding to boundary point $p_1^k$. For the first partition $FP_1$, the parameters $a_1$, $b_1$ are equal to the minimum value of the numeric attribute domain. For the last partition $FP_b$, the parameters $c_b$, $d_b$ are equal to the maximum value of the numeric attribute domain.
    Evaluate the information gain when each fuzzy partition is used to partition the numeric attribute domain for the current node.
    Return the fuzzy partition with the largest information gain.
end.
```

3.1.3 Selection of Branching Attribute

It is desirable to build small and correct trees in decision tree induction. The size of decision trees mainly depends on which attribute is selected for node branching in tree-building process. The attribute to be used for branching is called as *branching attribute*. To measure the quality of attributes as branching attribute, several measures have been developed[1]. The typical measures are the entropy measure, the gini index, Uuncertainty measure, and so on. The proposed decision tree induction method employs the entropy measure to measure the impurity of data set. The larger value the entropy measure gives, the more impure the data set is. Thus such measure is also called as the *impurity measure*. We prefer an attribute that generates less impure data sets in child nodes. The preference of attributes can be quantified by the measure called the *information gain* which is derived from the impurity measure.

Let S_k be a collection of data arrived at the node N_k: $S_k = \{(x, \mu_{S_k}(x)) | x \in U\}$ Here, x denotes a data and $\mu_{S_k}(x)$ denotes the membership degree of x to S_k. The entropy $Entropy(S_k)$ of S_k is defined as follows:

$$C_{S_k}^i = \sum_{x.class=i,\, x \in Supp(S_k)} \mu_{S_k}(x) \qquad C_{S_k} = \sum_i C_{S_k}^i$$

$$P_i^{S_k} = \frac{C_{S_k}^i}{C_{S_k}} \qquad Entropy(S_k) = -\sum_i P_i^{S_k} \log_2 P_i^{S_k}$$

The entropy measure is used to calculate the information gain which reflects the quality of an attribute as the branching attribute. As shown in Figure 4, let $S_k^{v|A}$ be a fuzzy subset of S_k such that $S_k^{v|A} = \{(x, \mu_{S_k^{v|A}}(x)) \mid \mu_{S_k^{v|A}}(x) = f(\mu_{S_k}(x), M(x.A, v)),\ x \in Supp(S_k)\}$. Here A denotes the branching attribute, and branching label v is either an element set for nominal attribute A or a trapezoidal fuzzy number for numeric attribute A. $M(x.A, v)$ denotes the matching degree of attribute A value of x to branching label v, and $f(\cdot, \cdot)$ is an aggregation function implemented by T-norm operators[14] like minimum, multiplication operation, etc. The entropy for $S_k^{v|A}$ is calculated like this:

$$Entropy(S^{k)}_{v|A} = -\sum_i P_i^{S^k_{v|A}} \log_2 P_i^{S^k_{v|A}}$$

The information gain of attribute A for data set S^k is defined as follows:

$$Gain(S^k, A) = Entropy(S^k) - \sum_{vinlterm(A)} \frac{C_{S_k^{v|A}}}{C_{S_k}} Entropy(S_k^{v|A})$$

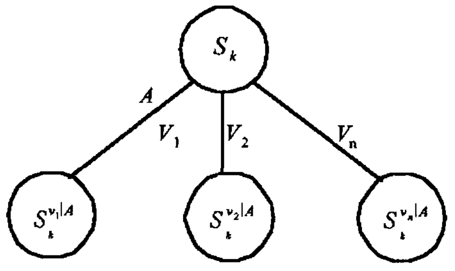


Fig. 4 Node expansion by the branching attribute A

As the information gain increases, the impurity of child nodes decreases. Thus the attribute that leads to the largest information gain, is selected as the branching attribute.

3.1.4 Branching Test At Each Branch

In the fuzzy decision trees, branches are labeled by either an element set for nominal attribute or a trapezoidal fuzzy number for numeric attribute. When a data follows down a branch, data is compared with the branch label to evaluate matching degree. For the evaluation of matching degree $M(\cdot, \cdot)$, the following matching measures are used: When attribute value is a numeric value c, the matching degree $M(c, L)$ of a numeric value c to trapezoidal fuzzy number L is defined by the membership degree of c to trapezoidal fuzzy number L like this:

$$M(c, L) = \mu_L(c)$$

In the case of nominal attributes, the matching degree $M(L, n)$ is determined by whether nominal attribute value n belongs to the label element set L like this:

$$M(L, n) = \begin{cases} 1 & \text{if } n \notin L \\ 0 & otherwise \end{cases}$$

3.1.5 Leaf Node Labeling

In fuzzy decision trees to be constructed, leaf nodes may represent multiple classes with different membership degrees. The proposed fuzzy decision tree building procedure uses the following class labeling method for leaf nodes: Let T_i be the data set arriving at terminal node N_i and $\mu_{T_i}(x)$ be the membership degree of data x to the set T_i. For each class j, we accumulate the membership degrees of data belonging to class j at node N_i as follows:

$$\delta_i^j = \sum_{x \in Supp(T^i),\, x.class = j} \mu_{T_i}(x)$$

The class label $CL(N_i)$ of leaf node N_i is assigned by the averaged representation degree $x_{N_i}(j)$ of node N_i for class k as follows:

$$CL(N_i)=\{\ (k, x_{N_i}(k))|\ \ k \in C,\ \ x_{N_i}(k)= \\ \delta_{N_i}^k | x|x \in Supp(T^i), x.class = j|\}$$

3.1.6 Procedure for Fuzzy Decision Tree Construction

The following shows the proposed procedure for fuzzy decision tree construction. In Step 1, we can control some parameters such as data threshold, branching factor, and maximally allowed impurity degree. The data threshold is used to ignore data whose membership degree is less than the threshold value in the course of tree-building. The branching factor is the maximum number of child nodes allowed for each node. The maximally allowed impurity degree is the quality of node at which no more node branching is needed.

```
procedure Fuzzy Decision Tree Construction
begin
    step 1. Set parameters for data threshold, branching factor,
            maximally allowed impurity degree, and make the root
            node.
    step 2. Place the initial data set on the root node and set
            the membership degree of each data to its class
            confidence degree.
    step 3. For each nominal attribute unused so far from the
            root to the current node, apply the nominal attribute
            value space partitioning procedure to determine the
            best partition for the nominal attribute.
    step 4. For each numeric attribute unused so far from the
            root to the current node,  apply the numeric attribute
            value space partitioning procedure to determine a good
            partition for the numeric attribute.
    step 5. Select the best attribute, which has the partitions to
            generate the least impure partitions, among all
            considered nominal and numeric attributes.
    step 6. Make child nodes according to the partitions for the
            selected attribute.
    step 7. Partition the current data set into some data subsets
            in reference to the branch labels and deliver the
            partitioned data subsets into its corresponding child
            nodes.
    step 8. To each newly generated child node, apply steps
            3-8 if all attributes have not yet been used on the path
            from the root node to the the child node and the
            impurity degree of the node is greater than the
            maximally allowed impurity degree.
end.
```

3.2 Transformation of Fuzzy Decision Trees into Fuzzy Classification Rules

In fuzzy decision trees, a path from the root node to a terminal node corresponds to a fuzzy classification rule. The internal nodes and link labels on such a path describe the condition (i.e., antecedents) of a classification rule, and the terminal node label represents the class to which data satisfying the conditions belong. Therefore it is straightforward to transform a fuzzy decision tree into a set of fuzzy classification rules.

The followings are the fuzzy classification rules for the fuzzy decision tree shown in Figure 2.

IF Job $\in A, D$ and Salary is *low* HEN class = $(c1, 0.4), (c3, 1.0)$.

IF Job $\in A, D$ and Salary is *high* and Age is *young* THEN class = $(c1, 1.0), (c3, 0.1)$.

IF Job $\in A, D$ and Salary is *high* and Age is *middle−aged* THEN class = {(c1,1.0), (c3,0.1)}.

IF Job is B THEN class = {(c1,1.0), (c2,0.2)}.

IF Job $\in A, C$ and Age is *middle−aged* THEN class = {(c2,1.0), (c3,0.3)}.

IF Job $\in A, D$ and age is *old* and Job change rate is *medium* THEN class = {(c2,1.0), (c3,0.2)}.

IF Job $\in A, D$ and age is *old* and Job change rate is *high* THEN class = {(c2,0.5), (c3,1.0)}.

3.3 Inference Procedure for Classification

The fuzzy classification rules learned by the proposed fuzzy decision tree induction method have crisp nominal predicates and fuzzy numeric predicates on rule antecedents and fuzzy set of classes on rule consequents. When such fuzzy classification rules are used to determine the class of a new data, a fuzzy inference method is applied. The fuzzy inference method makes each fuzzy classification rule determine the class of the data according to their rule. Then, it aggregates class results of each rule into one.

Each fuzzy classification rule R_i makes class decision for a new data x as follows: Calculate the matching degrees $pred_j(x)$ of data x to antecedent predicates $pred_j(\cdot)$ of rule R_i according to the method presented in Section 3.1.4. Combine the matching degrees $pred_j(x)$ into a rule satisfaction degree τ_i like this: Here T_j is a T-norm operator selected in the application domain.

$$\tau_i(x) = T_j(pred_j(x))$$

The classification rule R_i with the rule satisfaction degree $\tau_i(x)$ for data x determines the certainty factor with which data x belongs to class c as follows: Here $\mu_{Cons_i}(c)$ denotes the membership degree c to the consequent fuzzy set $Cons_i$

of rule R_i.

The classification rules of each rule are merged as follows: $cf(x, c)$ is the confidence degree with which data x belongs to class c.

$$cf(x, c) = \frac{x(x, c)}{\sum_i \tau_i(x)}$$

4. Experiments

To show the applicability of the proposed fuzzy classification rule learning method based on fuzzy decision tree induction, this section presents some experiment results. Table 1 shows the training data set of which data has two numeric attributes and class information with confidence degree. The proposed decision tree induction method produced the fuzzy decision tree shown in Figure 5. As we can see in the figure, the decision tree has membership functions as branching terms for numeric attributes, and it has terminal nodes which are labeled with a fuzzy set of classes (e.g., {(C1, 0.7), (C2, 0.8)}). From this fuzzy decision tree, fuzzy classification rules can be easily extracted. Suppose that linguistic terms are assigned for the membership functions used to partition the attribute domains as follows:

<u>attribute-1</u>

small=Trap(0, 0, 1.9, 2) *medium*=Trap(1.9, 2, 4, 5)
large=Trap(4, 5, 6.6)

<u>attribute-2</u>

low=Trap(0, 0, 2, 2.1) *moderate*=Trap(2, 2.1, 2,2, 3.1)
high=Trap(2.2, 3.1, 4.0, 4.0) *lower*=Trap(0, 0, 1.8, 2.1)
higher=Trap(1.8, 2.1, 4.0, 4.0)

For the fuzzy decision tree of Figure 5, the following fuzzy classification rules can be obtained with the fuzzy linguistic terms defined in the above.

IF attribute-1 is *small* and attribute-2 is *low* THEN class={(C2, 0.83)}.

IF attribute-1 is *small* and attribute-2 is moderate THEN class={(C1, 0.7), (C2, 0.8)}.

IF attribute-1 is *small* and attribute-2 is high THEN class={(C2, 0.95)}.

IF attribute-1 is *medium* THEN class={(C12, 1.0)}.

IF attribute-1 is *large* and attribute-2 is lower THEN class ={(C4, 0.88)}.

IF attribute-1 is *large* and attribute-2 is higher THEN class ={(C3, 1.0)}.

The following shows some classification results derived by the inference method of Section 3.3, where the left size values are test data and the right side sets indicate their corresponding class information.

(2,2) → {(C5,1)}
(1.95,2.0) → {(C1,0.42),(C5,0.5)}
(4.9,1.1) → {(C4,0.8),(C5,0.1)}

Table 1. Training Data I

ID	attribute-1	attribute-2	class	confidence degree
1	0	0.3	1	1
2	0	2	1	0.7
3	0.4	4	2	1
4	1	1	1	0.8
5	1	2.1	2	0.8
6	1.9	3.1	2	0.9
7	2	2	5	1
8	3	3	5	1
9	4	2	5	1
10	5	1	4	1
11	5.3	2.1	3	1
12	5	3	3	1
13	5.8	0.6	4	1
14	6.6	1.8	4	0.65
15	6	4	3	1
16	1.4	2.2	1	0.7

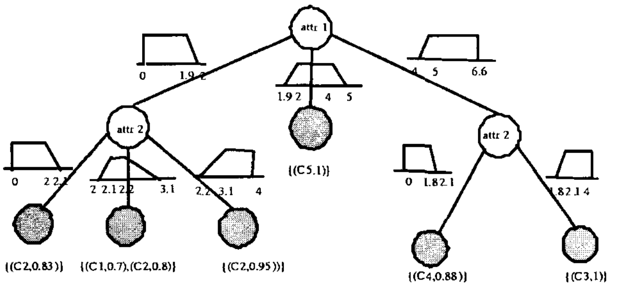


Fig.5 A fuzzy decision tree for the training data set of Table 1

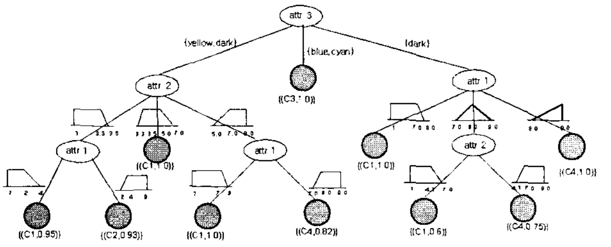


Fig.6 A fuzzy decision tree for the training data set of Table 2

Table 2. Training data II

	attribute-1	attribute-2	attribute-3	class	confidence degree
1	2	1	yellow	1	0.9
2	1	3	yellow	1	1
3	4	2	yellow	2	1
4	4.5	3	yellow	2	0.85
5	6	2	blue	3	1
6	7	1	blue	3	1
7	3	5	yellow	1	0.7
8	4	4.5	yellow	1	1
9	6	3.5	dark	1	1
10	8	4.1	dark	1	0.6
11	2	8	cyan	3	1
12	3.5	9	cyan	3	0.8
13	6	5	dark	1	1
14	7	9	dark	1	1
15	8	7	dark	4	0.75
16	9	9	dark	4	0.9

Table 2 shows another training data set of which data have two numeric attributes and one nominal attribute. The proposed decision tree induction method constructed a fuzzy decision tree shown in Figure 6. Since the branching factor of decision trees was set to 3 in the experiment, the number of partitions for each attribute (i.e., the number of child nodes for each node in the created decision tree) could not be greater than 3. Therefore the domain of a nominal attribute *attribute*-3 is partitioned into 3 sets of attribute values {*yellow, dark*}, {*blue, cyan*}, {*dark*}.

6. Conclusions

The rule-based systems are widely used for classification system implementation. This paper categorized the classification rules into 5 types according to the shapes of antecedents and consequents and examines their properties. Then it introduced a fuzzy classification rule learning method based on a fuzzy decision tree induction method.

The proposed method has the following characteristics: First, it can be applied to the training data set in which data can have both nominal attributes and numeric attributes and data class has a confidence degree on the interval [0,1]. Second, it generates a type of fuzzy classification rules belonging Type 3 classification rules, where antecedents contain fuzzy linguistic terms defined by trapezoidal fuzzy numbers for numeric attributes and sets of attribute values for nominal attributes. Third, it makes it possible to control the size of classification rules by setting the branching factor of decision trees to be constructed at will. Forth, it generates proper partitions (i.e., fuzzy linguistic terms for numeric attributes and sets of attribute values for nominal attributes) for attribute domains on the fly.

From the experiments, we have observed that the proposed method produces meaningful fuzzy classification rules. In order to improve the classification performance of fuzzy classification rules created by the proposed method, some fine-tuning method is needed. It is expected that genetic algorithm-based approach can be applied to the fine-tuning. This subject remains as a further study.

References

[1] T. M. Mitchell, Machine Learning, The McGraw-Hill Co., 414p, 1997.

[2] Y. Yuan, M. J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems, Vol.69, pp.125-139, 1995.

[3] A. Ittner, J. Zeidler, R. Rossius, W. Dilger, M. Schlosser, Feature space partitioning by non-linear and fuzzy decision trees, Proc. of the 7-th IFSA World Congress(Prague, Czech), pp.394-398, 1997.

[4] J. Zeidler, M. Schlosser, Continuous-valued attributes in fuzzy decision trees, Proc. of the 6-th Int. Conf. on Information, Processing and Management of Uncertainty in Knowledge-Based Systems (Granada, Spain), pp.395-400, 1996.

[5] D. Rasmussen, R. R. Yager, SummarySQL: A fuzzy tool for data mining, Intelligent Data Analysis, Vol.1, No.1, 1997.

[6] S. A. Maelainin, A. Bensaid, Fuzzy data mining query language, Proc. of the Second Int. Conf. on Knowledge-Based Systems(Adelaide, Australia), pp.335-340, 1998,

[7] Y. El-Sonbaty, M. A. Ismail, Fuzzy Clustering for Symbolic Data, IEEE Trans. on Fuzzy Systems, Vol.6, No.2, pp.195-204, 1998.

[8] C. Z. Janikow, Fuzzy Decision Trees: Issues and Methods, IEEE Trans. on Systems, Man, and Cybernetics - Part B, Vol.28, No.1, pp.1-14, 1998.

[9] K. Nozaki, H. Ishibuch, H. Tanaka, Adaptive Fuzzy Rule-based Classification Systems, IEEE Trans. on Fuzzy Systems, Vol.4, No.3, pp.238-250, 1996.

[10] R.L.P. Chang, T. Pavlidis, Fuzzy Decision Tree Algorithms, IEEE Trans. on Systems, Man, and Cybernetics, Vol.7, No.1, pp.28-35, 1977.

[11] T. Tani, M. Sakoda, Fuzzy modeling by ID3 algorithm and its application to prediction of heater outlet temperature, Proc. of IEEE Int. Conf. on Fuzzy Systems (San Diego), pp.923-930, 1992.

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

[13] J. Jang, Structure determination in fuzzy modeling: A fuzzy CART approach, Proc. IEEE Conf. on Fuzzy Systems, pp.480-485, 1994.

[14] H.-J. Zimmermann, Fuzzy Set Theory and its Applications, Kluwer Academic Pub., 399p, 1991.

Keon Myung Lee

He received B.S., M.S., and Ph.D. in Dept. of Computer Science from Korea Advanced Institute of Science and Technology (KAIST) in 1990, 1992, and 1995, respectively. He joined Chungbuk National University, Korea in 1996 and is currently an associate professor at School of Electric and Computer Engineering. His research interests include data mining, agent systems, soft computing, bioinformatics, and computer security.

Hak Joon Kim

He received B.S. in Dept. of Mathematics Education from Seoul National University, and received M.S. in Dept. of Computer Engineering from Songsil University. He had worked for DACOM from 1985 to 1997. He joined Howon University in 1997 as an assistant professor. He is pursuing Ph.D. in Dept. of Computer Science, Chungbuk National University from 2001. His research interests include artificial intelligence, and software engineering.