A Fuzzy Decision Tree Induction Method for Fuzzy Data

Koen-Myung Lee†, Kyung-Mi Lee†, Jee-Hyong Lee§, Hyung Lee-Kwang §
†Dept. of Comp. Science, Chungbuk Nat. Univ., CheongJu ChungBuk, 361-763, Korea
§School of Comp. Science and Syst. Eng., Kyushu Inst. of Tech., Fukuoka, 820, Japan
§Dept. of Comp. Science, KAIST, Thejon, 305-701, Korea
(e-mail: kmlee@cbucc.chungbuk.ac.kr)

Abstract

Decision tree induction is one of useful approaches for extracting classification knowledge from a set of feature-based examples. Due to observation error, uncertainty, subjective judgement, and so on, many data occurring in real world are obtained in fuzzy description. Although several fuzzy decision tree induction methods have been developed for fuzzy data, they are improper to deal with some types of fuzzy data. This paper proposes a fuzzy decision tree induction method for fuzzy data of which numeric attributes are represented by fuzzy number, interval value as well as crisp value, of which nominal attributes are represented by crisp nominal value, and of which class has confidence factor. It presents a tree construction procedure to build a fuzzy decision tree from a collection of fuzzy data and an inference procedure for fuzzy decision tree to classify new fuzzy data. It also presents some experiment results to show the applicability of the proposed method.

1. Introduction

Decision tree induction has been widely used in extracting knowledge from feature-based examples for classification and decision making[1]. Many methods have been developed to construct decision trees from collection of data. Due to observation error, uncertainty, subjective judgement, and so on, many data occurring in real world are obtained in fuzzy description. In the classification field, some fuzzy decision tree induction methods are required to deal with such fuzzy data. In the literature, we can find several fuzzy decision tree induction methods which produce some fuzzy decision tree to handle some aspect of fuzziness of data[2,3,4,8,10,12]. According to the type of training data, they can be classified into two categories as follows: One is the induction methods which construct fuzzy decision tree from a collection of crisp data. Their decision trees use fuzzy linguistic terms to improve generalization property. The use of fuzzy linguistic terms makes it possible to avoid sudden and inappropriate class changes by small changes in the attribute values of data being classified. The other is the induction methods which construct fuzzy decision trees from a collection of fuzzy data. The researches on fuzzy decision tree induction for fuzzy data have not yet sufficiently performed. The existing induction methods are improper to deal with some types of fuzzy data. We are interested in fuzzy decision decision tree induction for fuzzy data, not yet studied, of which numeric attributes are represented by fuzzy number, interval value as well as crisp value, of which nominal attributes are represented by crisp nominal value, and of which class has confidence factor.

This paper is concerned with a fuzzy decision tree induction method for such fuzzy data. It proposes a tree-building procedure to construct fuzzy decision tree from a collection of fuzzy data and an inference procedure for fuzzy decision tree to determine class of new data.

This paper is organized as follows: Section 2 briefly introduces data representation considered in this study. Section 3 surveys several existing fuzzy decision trees and presents the characteristics of the fuzzy decision tree to be considered in this study. Section 4 presents the proposed fuzzy decision tree building procedure and inference procedure for classification. Section 5 shows some experiment results and finally Section 6 draws conclusions.

2. Data Representation

A data $D_i$ of an object or an event is represented with a set of attributes describing the object like this: Here $A_i^j$ denotes the $r$-th attribute for the object.

$$D_i = (A_i^1, A_i^2, ..., A_i^n)$$

Some attributes have discrete nominal domain, and others have continuous numeric domain. In this study, we assume that discrete nominal domain attributes are characterized by crisp values and continuous numeric domain attributes are characterized by crisp values, interval values and fuzzy numbers. In the case of training data, each data has the class information along with its confidence degree.

In description of fuzzy values for continuous numeric attributes, trapezoidal fuzzy numbers are widely used since they can sufficiently well represent fuzzy values and they are simple to describe and process[14]. Trapezoidal fuzzy numbers $\text{Trap}(\alpha, \beta, \gamma, \delta)$ are defined as follows:

$$\text{Trap}(\alpha, \beta, \gamma, \delta) = \begin{cases} 0 & \text{if } x < \alpha \\ (x - \alpha)/(\beta - \alpha) & \text{if } \alpha \leq x \leq \beta \\ 1 & \text{if } \beta \leq x \leq \gamma \\ (x - \gamma)/(\delta - \gamma) & \text{if } \gamma \leq x \leq \delta \\ 0 & \text{if } x > \delta \end{cases}$$

In this study we assume that trapezoidal fuzzy numbers are used to represent fuzzy values. Under this assumption, all continuous numeric attribute values are represented with trapezoidal fuzzy numbers as follows[14]:

\[\text{Trap}(\alpha, \beta, \gamma, \delta)\]

The authors wish to acknowledge the financial support of the Korea Research Foundation made in the program of 1998-1999.
3. 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 domain data, and CART is designed to deal with continuous numeric domain data. A number of alternation of them have been reported in the literature. Fuzzy decision trees are one of them.

In classical decision trees, nodes make a 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 condition of nodes.

Several fuzzy decision tree construction methods have been proposed so far[2][4][8][10][12]. In [11][13], decision tree construction methods are incorporated into fuzzy modeling. They use the decision tree building methods to determine effective branching attributes and their splitting intervals for classification of crisp data. These intervals are then used to determine fuzzy boundaries for input variables, which will be used to form fuzzy rules. As a matter of fact, they use the decision tree construction methods for preprocessing and not for building fuzzy decision tree. In [2], the fuzzy decision tree construction method is designed for classification problem with attributes and classes represented in fuzzy linguistic terms, in fact, fuzzy membership function. When it builds fuzzy decision trees, it uses predefined fuzzy linguistic terms with which the attribute values of training data are fuzzified. In [4], a decision tree induction method is proposed to construct fuzzy decision trees from a collection of crisp data of which all attributes are continuous numeric one and their values are crisp. In [8], a decision tree induction method is developed to generate fuzzy decision trees which can be used for both classification and regression. Its training data may contain fuzzy values in somewhat restricted form, but fuzzy linguistic terms used in fuzzy decision tree should be predefined by experts or other preprocessing procedures. Even though several fuzzy decision tree induction methods have been proposed, they cannot always be applied to all types of fuzzy data.

In this paper, we propose a fuzzy decision tree induction method for data of which discrete nominal attributes have crisp value, and of which continuous numeric attributes have crisp value, interval value, and fuzzy number. From now on, we will use the term fuzzy data to refer to such data.

The proposed fuzzy decision tree induction method has the following properties: First, its training data are fuzzy data of which nominal attributes are represented by crisp nominal value and of which continuous numeric attributes are represented by fuzzy number, interval value, or crisp value, 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]. After constructing tree, we may assign descriptive linguistic terms to the generated fuzzy membership functions so as to enhance comprehensibility. Third, the tree induction method can control the branching factor at will. For this, nodes corresponding to nominal attribute have sets of nominal values in their branches. Figure 1 exemplifies a fuzzy decision tree considered in this study. The root node corresponds to 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 determined 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 is aggregated.

4. The Proposed Fuzzy Decision Tree

Fuzzy decision tree induction has 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] which 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 selec-
tion method, branch 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.

4.1. Attribute Value Space Partitioning

To avoid the explosive growth of fuzzy decision tree, the branching factor of the constructed decision tree should be controllable. It means that the fuzzy decision tree building procedure should have the ability to partition attribute value spaces into any desired number of partitions. The proposed fuzzy decision tree building procedure partitions nominal attribute value spaces with sets of nominal attribute values, and continuous numeric attribute value spaces with trapezoidal fuzzy numbers.

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

$$CWS(A) = \{D_j, A | D_j.class = i, \mu_{N_k}(D_j) > 0\}$$

Here $CWS(A)$ 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:

$$CWS(A) = \{a, b, c\}, CWS^2(A) = \{b, c, d\}, CWS^3(A) = \{c, e\}$$

$$CWS(A) = \{d, e\}$$

The classwise element set $CWS(A)$ implies that data whose attribute $A$ has a value in $CWS(A)$ at node $N_k$ probably belongs to class $i$. In this perspective, we use classwise element sets $CWS(A)$ 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, we can consider $b^c \div b!$ possible partitioning combinations based on $CWS(A)$ since it is the same situation to divide $c$ elements into $b$ groups. For example, in the above, the partitioning combination $\{CWS^1(A), CWS^2(A)\}$ and $\{CWS^3(A), CWS^4(A)\}$ 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 for nominal attribute space partitioning**

- Find the classwise element sets $CWS$ for the current attribute.

- if (the size of classes $\geq$ the branching factor) then

  Generate all possible partitioning combinations from $CWS$.

  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.

  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.

Figure 2: Numeric attribute value space partitioning

For continuous numeric attribute value space partitioning, we use trapezoidal fuzzy numbers to form fuzzy partitions. We can make 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 continuous numeric attribute value space. In our study, all continuous numeric attribute values can be represented by trapezoidal fuzzy numbers as mentioned in Section 2. Let us see the endpoints (we call it boundary points) of the supports of trapezoidal fuzzy numbers to represent attribute values. It is reasonable to locate fuzzy partitioning boundaries between adjacent two boundary points belonging to different classes each other. We take this idea in constructing fuzzy linguistic terms for continuous numeric attribute value space partitioning. Let us see Figure 2. Figure 2-(a) shows fuzzy values for a numeric attribute, which are represented by their membership functions. Figure 2-(b) shows boundary points extracted from the supports of the fuzzy values. Each boundary point is labeled with the
class to which the data with the boundary point belongs. We locate fuzzy partitioning boundaries between these boundary points. Meanwhile, it is undesirable to locate fuzzy partitioning boundary between boundary points belonging to the same class like between \( P_3 \) and \( P_5 \). To alleviate this problem, we remove boundary points, like \( P_3 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear in- 
side the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belonging to the same class. If we partition the attribute value space with 4 fuzzy linguistic terms, we randomly choose just preceding boundary points, like \( P_3 \) just preceding \( P_5 \), which appear inside the subsequences of boundary points belongin...
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:

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

4.4. Leaf Node Labeling

In our fuzzy decision trees, 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 $S_k$ be the data set arriving at node $N_k$ and $\mu_{S_k}(x)$ be the membership degree of data $x$ to the set $S_k$. For each class $c$, we accumulate the membership degrees of data belonging to class $c$ at node $N_k$ as follows:

$$\delta_{N_k}^c = \sum_{x \in \text{Supp}(S_k), x \text{ class}=c} \mu_{S_k}(x)$$

The class label $CL(N_k)$ of leaf node $N_k$ is assigned by the normalized representation degree $\kappa_{N_k}(c)$ of node $N_k$ for class $c$ as follows:

$$CL(N_k) = \{(c, \kappa_{N_k}(c)) | c \in C\}$$

$$\kappa_{N_k}(c) = \frac{\delta_{N_k}^c}{\max_c \delta_{N_k}^c}$$

4.5. 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 for Fuzzy Decision Tree Construction**

1. Set parameters for data threshold, branching factor, maximally allowed impurity degree.
2. Place the initial data set on the root node and set the membership degree of each data to its class confidence degree.
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.
4. For each numeric attribute unused so far from the root to the current node, apply the continuous numeric attribute value space partitioning procedure to determine the best partition for the numeric attribute.
5. Select the best attribute among all considered nominal and numeric attributes.
6. Make child nodes according to the partition by the selected attribute.
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.
8. To each newly generated child node, apply steps 3–8 if all attributes have not yet been used and the impurity degree of the node is greater than the maximally allowed impurity degree.

**end.**

4.6. Inference Procedure for Decision Making

Once a fuzzy decision tree is built for some training data set, the tree can be used to classify new data. To determine the class of new data, fuzzy decision tree performs a kind of inference as follows: New data is fed into the root node with its initial membership degree 1. Whenever a node $N_k$ receives a data $x$, it evaluates the matching degree $M(x.A, v)$ of the data attribute value $x.A$ to be tested with its branch labels $v$. The membership degree of data $x$ to each child node is then calculated by $f(\mu_{N_k}(x), M(x.A, v))$, where $\mu_{N_k}(x)$ is the membership degree of data $x$ to node $N_k$, and $f(\cdot, \cdot)$ is a T-norm operator [14]. When the calculated membership degree is greater than the pre-specified data threshold, it passes the data into the corresponding child node with the membership degree. When data attribute value is missing for attribute to be tested, it passes the data into all its child nodes by assuming that the data is perfectly matched with branch labels to be tested.

In fuzzy decision trees, data fed into the root node may arrive at several leaf nodes and leaf nodes are usually labeled by multiple classes with different membership degrees. So as to assign class to new data, we use the following method: Here $\kappa_{N_k}(c)$ is the representation degree of node $N_k$ for class $c$, $\mu_{N_k}(x)$ is the membership degree of data $x$ to $S_k$ which is the set of data arriving at leaf node $N_k$, and $cf(x, c)$ is the confidence degree with which data $x$ belongs to class $c$.

$$K(x, i) = \sum_{N_k \in \text{leaf nodes}} \kappa_{N_k}(i) \cdot \mu_{N_k}(x)$$

$$cf(x, c) = \frac{K(x, c)}{\max_i K(x, i)}$$

5. Experiments

To show the applicability of the proposed fuzzy decision tree, we performed several experiments. The following shows one experiment result. Table 1 shows the training data set of which data has two numeric attributes. [a, b] indicates the interval value ranged from a to b, and Trap(a, b, c, d) indicates a trapezoidal fuzzy number. Each training data is assumed to have its class confidence degree 1.

For the training data of Table 1, the proposed tree construction procedure generated the fuzzy decision tree shown in Figure 4. In the figure, trapezoidal fuzzy numbers correspond to fuzzy linguistic terms for branches and the fuzzy sets of leaf nodes describe the confidence degree with which data arriving at leaf node belongs to classes.

The following shows the classification results for some data by the built fuzzy decision tree.

- **data**: (Trap(1.0,1.2,1.4,1.6), Trap(1.0,1.2,1.4,1.6))
- **class**: (1.0,0),(2.0,0),(3.0,0),(4.0,0)
- **data**: (Trap(4.2,4.6,5.3,6.0), Trap(3.0,3.2,3.4,3.6))
Table 1: Training data

<table>
<thead>
<tr>
<th>ID</th>
<th>attribute 1</th>
<th>attribute 2</th>
<th>class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>[0.0,0.5]</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Trap(0.2,0.4,0.5)</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>[1.8,2.3]</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>(1.0,1.2)</td>
<td>(3.0,3.3)</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>Trap(1.8,2.0,2.4,2.6)</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Trap(4.8,4.9,5.3,5.4)</td>
<td>Trap(1.6,1.7,2.0,2.3)</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>[5.7,6.2]</td>
<td>Trap(0.0,2.0,5.0,6)</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>[6.0,6.6]</td>
<td>[1.8,2.3]</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>6</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>Trap(1.2,1.3,1.4,1.4)</td>
<td>Trap(1.9,2.0,2.1,2.2)</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>Trap(5.1,5.2,5.4,5.5)</td>
<td>Trap(1.9,2.0,2.2,2.2)</td>
<td>3</td>
</tr>
</tbody>
</table>

We can see that the built fuzzy decision tree and classification results for new data are consistent with our intuition.

6. Conclusions

This paper proposed a fuzzy decision tree induction method for fuzzy data. The proposed fuzzy decision tree induction method can be applied to fuzzy data whose nominal attributes are represented by crisp values and continuous numeric attributes are represented by fuzzy numbers, fuzzy interval, and crisp value, and of which class is represented by a nominal value with confidence degree ranged on the interval $[0,1]$. The decision tree-building procedure generates trapezoidal fuzzy numbers for continuous attributes during its process. Thus, it is unnecessary to predetermine trapezoidal fuzzy numbers before applying the tree-building procedure. After constructing a decision tree, we can assign descriptive linguistic terms to generated trapezoidal fuzzy numbers so as to enhance its comprehensibility. In the tree-building procedure, we can control tree branching factor at will. In addition, we introduce a method for nominal attribute value space partitioning. The proposed inference procedure allows data with missing attribute value to be classified by fuzzy decision tree. From the experiments, we have observed that the proposed fuzzy decision tree induction method produces meaningful results. In this study, we have not yet considered the tree node pruning for the purpose of generalization. Some more studies are still needed for this subject.

References