Learning a Fuzzy Decision Tree from Uncertain Data

Hang Yu
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney
Sydney, Australia
Hang.Yu@student.uts.edu.au

Jie Lu
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney
Sydney, Australia
Jie.Lu@uts.edu.au

Guangquan Zhang
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney
Sydney, Australia
Guangquan.Zhang@uts.edu.au

***Abstract*—Uncertainty in data exists when the value of a data item is not a precise value, but rather by an interval data with a probability distribution function, or a probability distribution of multiple values. Since there are intrinsic differences between uncertain and certain data, it is difficult to deal with uncertain data using traditional classification algorithms. Therefore, in this paper, we propose a fuzzy decision tree algorithm based on a classical ID3 algorithm, it integrates fuzzy set theory and ID3 to overcome the uncertain data classification problem. Besides, we propose a discretization algorithm that enables our proposed Fuzzy-ID3 algorithm to handle the interval data. Experimental results show that our Fuzzy-ID3 algorithm is a practical and robust solution to the problem of uncertain data classification and that it performs better than some of the existing algorithms.**

Keywords-uncertain data classification; fuzzy decision tree; fuzzy set theory; ID3; discretization method; interval data

I. INTRODUCTION

Classification is one of the key processes in the data mining area and has significant application merits in many fields. In traditional classification, the value of data is precise and definite. However, due to new techniques in data acquisition, a large amount of data with imprecise values is now generated and collected for use in many real-world applications, such as wireless sensor networks [1], moving object detection [2], and mobile telecommunications [3,4]. Since there are intrinsic differences between uncertain and deterministic data, it is difficult to deal with uncertain data using traditional classification algorithms.

Hence, some existing classification methods only consider the uncertainty of attributes and assume the class type is certain. This assumption is convenient, but not reflective of reality, because in many practical situations an object's class labels can also be uncertain. In other words, it can also be represented by a probability distribution of multiple values. This paper focuses on an uncertain data classification problem and represent uncertain attributes and class labels for objects using uncertain models. To solve the problem, we firstly propose a discretization method that can convert interval data into a probability distribution. Then a fuzzy ID3 algorithm is presented to solve the uncertain data classification problem. Fuzzy ID3 is an extension of the existing ID3 algorithm; it integrates fuzzy set theory and ID3 to overcome the effects of spurious precision in the data, to treat uncertainties in the data and to reduce the decision tree sensitivity to small changes in attribute values.

In the remainder of this paper, some related works are briefly described in Section 2. The definition of the uncertain data classification problem is given in Section 3. Section 4 presents our discretization method, and our Fuzzy-ID3 algorithm is presented in Section 5. The experimental studies on performance are presented in Section 6, and we present our conclusion in Section 7.

II. RELATED WORK

Several data classification algorithms have been used to classify uncertain data, such as support vector machines (SVM) algorithm [6], extreme learning machines (ELM) algorithm [7], Bayesian classification algorithms [8], rule-based classification algorithms [9], decision tree algorithms [10], and so on. Of these algorithms, decision tree algorithms are popular because they are practical and easy to understand. However, the decision tree sensitivity to small changes in attribute values.

Fuzzy set theory offers a rich spectrum of methods for the management of uncertainty [12]. Therefore, some researchers integrates fuzzy set theory into the decision tree, such as fuzzy extension of ID3 [11] and trees based on fuzzy rules [13]. In these models, a node on a decision tree does not use a crisp test to deterministically decide which branch to send a training or testing object down. Rather, a fuzzy test is administered on a point-valued object [15]. Each branch from root to leaf can be converted into a rule with a condition.

However, most of decision tree algorithms only handle category attributes, hence the numeric attributes need be partitioned into a number of sub-ranges and treat each such sub-range as a category. This process of partitioning numeric attributes into categories is usually termed discretization [17]. However, uncertain numeric attribute's value is often represented by an interval with a probability distribution function over the interval. Thus, most of newest discretization methods cannot work. Such as based on CIAM [17], information entropy [18], interval Similarity [19].

III. PROBLEM DEFINITION

Normally, uncertain models of attributes can be classified into two categories [5-14]: categorical and numerical. In numerical models, the i-th uncertain numerical attribute

(UNA) is denoted by $A_i^{\mu_n}$. The value of $A_i^{\mu_n}$ is represented as an interval and a PDF over this interval. The j-th object of $A_i^{\mu_n}$, denoted by $A_{ij}^{\mu_n}$, is an interval $[A_{ij}^{\mu_n}.l,\ A_{ij}^{\mu_n}.r]$ where $A_{ij}^{\mu_n}.l,\ A_{ij}^{\mu_n}.r\ \in R,\ A_{ij}^{\mu_n}.r \geq A_{ij}^{\mu_n}.l$. The uncertain PDF of $A_{ij}^{\mu_n}$, denoted by $A_{ij}^{\mu_n}.f(x)$, is a probability distribution function of $A_{ij}^{\mu_n}$, such that $\int_{A_{ij}^{\mu_n}.l}^{A_{ij}^{\mu_n}.r} A_{ij}^{\mu_n}.f(x)dx = 1$. In categorical models, the i-th uncertain categorical attribute (UCA) is denoted by $A_i^{\mu_c}$. The j-th object of $A_i^{\mu_c}$, denoted by $A_{ij}^{\mu_c}$, takes values from the categorical domain Dom with cardinality $|Dom| = n$ and $Dom = \{d_1, \dots, d_k\}$. The i-th UCA $A_i^{\mu_c}$ is characterized by its probability distribution over Dom and can be represented by the probability vector $P = \{p_1, \dots, p_k\}$ such that $P(A_{ij}^{\mu_c} = d_i) = p_i$ and $\sum_{i=1}^{k} p_i = 1\ (1 \leq i \leq k)$.

The class label also can be represented in uncertain models. The i-th uncertain class label (UCL) is denoted by C_i^{u}. The j-th object of C_i^{u}, denoted by C_{ij}^{u}, takes values from the class domain Dom with cardinality $|Dom| = n$and $Dom = \{c_1, \dots, c_k\}$. The UCL of the j-th object is characterized by its probability distribution over Dom and can be represented by the probability vector $P = \{p_1, \dots, p_k\}$ such that $P(A_{ij}^{\mu_c} = c_k) = p_i$ and $\sum_{i=1}^{k} p_i\ = 1\ (1 \leq i \leq k)$.

TABLE I. A SEQUENCE OF UNCERTAIN SAMPLES

Case	*Age*	*Occupation*	*Location*	*Products*
1	20-30	Singer:0.9 Doctor:0.1	Online:0.8 Supermarket:0.2	Coffee:0.8 Wine:0.2
2	22-32	Singer:0.7 Artist:0.3	Online:0.1 Supermarket:0.9	Book:0.4 Coffee:0.6
3	27-37	Singer:0.3 Doctor:0.7	Online:0.2 Supermarket:0.8	Book:0.9 Coffee:0.1
4	23-33	Doctor:0.1 Artist:0.9	Online:0.5 Supermarket:0.5	Wine:1.0
5	37-47	Doctor:0.7 Artist:0.3	Online:0.7 Supermarket:0.3	Book:0.2 Wine:0.8
6	40-50	Doctor:0.3 Artist:0.7	Online:0.0 Supermarket:1.0	Wine: 1.0
7	38-48	Doctor:1.0	Online:0.2 Supermarket:0.8	Book:0.7 Wine:0.3
8	20-30	Singer:1.0	Online:0.6 Supermarket:0.4	Book:0.2 Coffee:0.8
9	37-47	Singer:0.9 Doctor:0.1	Supermarket:1.0	Coffee:0.3 Wine:0.7
10	20-30	Singer:0.7 Doctor:0.3	Online:1.0	Book:0.3 Coffee:0.7

Table I shows an example of our proposed uncertain data classification problem. “Age” is an UNA, while “Occupation” and “Location” are UCAs. The class label, or UCL, represent the possibility of purchasing a product. For instance, the class label for Case 1 mean that the possibility of purchasing “Coffee” is greater than “Wine”.

IV. A DISCRETIZATION METHOD

As described earlier, the value of an UNA is an interval-value data with an associated PDF. For example, in Table 1, “Age” is an UNA, whose precise value is not available. We only know the range of the attribute and the $PDF\ f(x)$ over that range. However, decision tree algorithms carry out a selection process of categorical attributes and cannot handle continuous ones directly [16]. Hence, a discretization method is needed to convert numerical attributes into categorical attributes. But most of newest discretization methods cannot handle interval-valued data. Hence in this paper, we propose a discretization method for convert UNA to UCA. Such as the UNA “Age [23-33]” can be represented by a possibility distribution P = {0.7($-\infty - 30$), 0.3(30-40), 0.0(40-+∞)}.

Hence, there are two key problems to be tackled in our discretization method. One is how to get the categorical domain Dom by an UNA attribute. The other is how to calculate the probability distribution over the Dom.

A. Get Categorical Domain

The main idea of our method is presented as follows: Let $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ denote two adjacent objects of $A_i^{\mu_n}$. If the class label of $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are not same, at the same time $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are also not similar, a boundary point of the variables d_i in the categorical domain Dom exist between $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$. However, if only the class label of $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are not same, whether or not a boundary point exists between $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ cannot be determined. Because if $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are very similar, the different class label between $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ may be due to other attributes.

The definition of the similarity measure of the two intervals $A = [a^-, a^+]$ and $B = [b^-, b^+]$ follows.

Definition 1. The similarity degree is the ratio S_V ($A_{ij}^{\mu_n}$, $A_{ij+1}^{\mu_n}$) which takes values from the interval [0, 1].

$$S_V(A,B) = \frac{1}{2}\left(\frac{\vartheta|A \cap B|}{\omega|A|} + \frac{\vartheta|A \cap B|}{\omega|B|}\right) \tag{1}$$

where $\vartheta[A \cap B]$ represents the distance of A and B, defined as:

$$\vartheta[A \cap B] = max[0, (\max[a^-, b^-] - min[a^+, b^+])] \tag{2}$$

and $\omega[X]$ represents the length of an interval X, defined as:

$$\omega[X] = upperbound(x) - lowerbound(x) \tag{3}$$

In our discretization method, due to the class label of one objects of $A_i^{\mu_c}$ be represented by the probability vector $P = \{p_1, \dots, p_k\}$ such that $P(A_{ij}^{\mu_c} = V_k^c) = p_i$, hence If the class label of $A_{ij}^{\mu_c}$ and $A_{ij+1}^{\mu_c}$ are same means:

$$P_{max}\left(A_{ij}^{\mu_c} = V_k^c\right) = P_{max}\left(A_{ij+1}^{\mu_c} = V_k^c\right) \tag{4}$$

where $P_{max}()$ represents the max probability among the probability vector $P = \{p_1, \dots, p_k\}$.

The discrimination process consists of the following steps:

Step1: Because each instance of an UNA has a maximal value and a minimal value, called critical points, we can place all the instances in ascending of their critical points.

Step2: Scanning the entire instance set in ascending order, let $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ denote two adjacent instances:

- If $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are same class label, then keep scanning.
- If $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are not same class label, but S_V $(A_{ij}^{\mu_n}, A_{ij+1}^{\mu_n}) > \theta$, then keep scanning.
- If $A_{ij}^{\mu_n}$ and $A_{ij+1}^{\mu_n}$ are not same class label, but S_V $(A_{ij}^{\mu_n}, A_{ij+1}^{\mu_n}) < \theta$ then set a break stop T_i = max ($A_{ij}^{\mu_n}$.r, $A_{ij+1}^{\mu_n}$.l) and keep scanning.

Step 3: T = { T_1, T_2, ..., T_n} is the permutation of the break stops, sorted so that $T_i \le T_{i+1}$ for i = 1,..., n. Then, a *Dom D* = {(-∞, T_1],…, (T_i, T_{i+1}],…, (T_n, +∞)} and i = 1,..., n-1.

B. Calculating the Probability Distribution

Once the *Dom* is constructed, the next thing is to measure the probability of a variable d_i. Assume an instance of interval attribute $A_{ij}^{\mu_n}$ has an interval value of [$A_{ij}^{\mu_n}$.*l*, $A_{ij}^{\mu_n}$.*r*] and a *PDF* of *f*(*x*) over that range. The possibility $p_i(x)$ of the variable d_i = (T_i, T_{i+1}] can be measured by:

$$\begin{cases} 0, & A_{ij}^{\mu_n}.r < T_i \ or \ T_{i+1} \le A_{ij}^{\mu_n}.l \\ \int_{T_i}^{A_{ij}^{\mu_n}.r} f(x)dx, & A_{ij}^{\mu_n}.l < T_i \le A_{ij}^{\mu_n}.r < T_{i+1} \\ \int_{A_{ij}^{\mu_n}.l}^{T_{i+1}} f(x)dx, & T_i < A_{ij}^{\mu_n}.l < T_{i+1} \le A_{ij}^{\mu_n}.r \\ \int_{T_i}^{T_{i+1}} f(x)dx, & A_{ij}^{\mu_n}.l < T_i < T_{i+1} \le A_{ij+1}^{\mu_n}.r \\ 1, & T_i < A_{ij}^{\mu_n}.l < A_{ij}^{\mu_n}.r \le T_{i+1} \end{cases} \quad (5)$$

Therefore, the UNA can be converted into UNC by our discretization method. Using the object "[23-33]" of attribute "Age" in Table 1 as an example, the *Dom D* of object "[23-33]" is {(−∞, 30], (30, 40], (40, +∞)} by our discretization. The probability vector P of object "[23-33]" is {0.7, 0.3, 0.0}. Hence the UNAs and UNCs can be represented in same terms, and our fuzzy decision tree algorithm is able to ignore the differences between the two kinds of data.

V. The Fuzzy ID3 Algorithm

The fuzzy set theory was given by Lotfi Zadeh [22] and is a good way to represent uncertainty. In this theory [23], the uncertainty can be presented by a membership. Normally, a fuzzy set F [24] in a universe of discourse U is characterized by a membership function μ_F, which takes values from the interval "[0,1]". For example, $u \in U$, $\mu_F(u) = 1$ means that u is definitely a member of F, $\mu_F(u) = 0$ means that u is definitely not a member of F, and $0 < \mu_F(u) < 1$ means that u is partially a member of F. Therefore, the possibility of taking value x for Y among all elements in X can be interpreted as a fuzzy membership function μ_F of a fuzzy variable Y defined on X [25].

In this paper, we propose a Fuzzy-ID3 algorithm. Like most fuzzy decision tree algorithms, our proposed algorithm for constructing decision trees also take a top-down approach, by choosing the attributes at each step that best splits the set of items. When the stopping condition is reached, the decision tree stops growing.

A. Fuzzy Set Theory

Definition 1. Let F_1 and F_2 be two fuzzy sets in U with the respective membership functions μ_{F_1} and μ_{F_2}. The union $F_1 \cup F_2$ is defined for all $u \in U$ by $\mu_{F_1 \cup F_2}(\mu) = max\{\mu_{F_1}(u), \mu_{F_2}(u)\}$. The intersection $F_1 \cap F_2$ is defined by $\mu_{F_1 \cap F_2}(\mu) = min\{\mu_{F_1}(u), \mu_{F_2}(u)\}$.

Definition 2. The cardinality measure (or sigma count) of a fuzzy set F is defined by $M(F) = \sum_{u \in U} \mu_F(u)$, which is the measure of the size of F.

B. Selection Criteria

In fuzzy decision tree algorithm, when given a dataset D, a high quality selection criteria is to select the attribute that most reduces the classification uncertainty [20]. The smaller classification uncertainty of a node means the more data's class label are same. In most of fuzzy ID3 algorithms, the information entropy is a measure classification uncertainty [21]. It reaches its minimum (zero) when all the cases in a node fall into a single target category. The smaller the information entropy means the smaller classification uncertainty. Therefore, in our fuzzy ID3 algorithm, the selection criteria is to select the attribute with minimum information entropy at each step.

However, in the decision tree construction process, an attribute A could be a root node, or an inner node, that is connected to the branch of its parent node. Figure 1 gives an example of attribute A is a root node. Figure 2 gives an example of attribute A is an inner node. Therefore, the classification uncertainty of attribute A can be calculated in one of two ways.

Definition 3. When an attribute A is a root node, its classification uncertainty is defined with the class label $\mathrm{C} = \{C_1, \ldots, C_J\}$ as follows:

$$C_u(A, L) = \frac{1}{n} \sum_{i=1}^{n} C_u(L_i) \quad (6)$$

where L_i is i-th linguistic value of attribute A. For example, in a linguistic value L_i of the attribute "Occupation" could be "Doctor". $C_u(L_i)$ is the classification uncertainty of the i-th linguistic value, defined as:

$$C_u(L_i) = -\sum_{j=1}^{J} \pi^*(C_j | L_i) \log_2 \pi^*(C_j | L_i) \quad (7)$$

where $\pi^*(C_j | L_i)$ is the normalization of the $\pi(C_j | L_i)$, defined as:

$$\pi_i^*(C_j|L_i) = \frac{\sum_{j=1}^{J} \pi(C_j|L_i)}{\sum_{i=1}^{n}\sum_{j=1}^{J} \pi(C_j|L_i)} \quad (8)$$

where $\pi(C_j|L_i)$ represents the represents the degree of truth of the classification rule: "IF L_i THEN C_j", which can be defined as:

$$\pi(C_j|L_i) = \frac{M(L_i \cap C_i)}{M(L_i \cup C_i)} = \frac{\sum_{u \in U} min\left(\mu_{L_i}(u), \mu_{C_i}(u)\right)}{\sum_{u \in U} max\left(\mu_{L_i}(u), \mu_{C_i}(u)\right)} \quad (9)$$

Definition 8. When an attribute A is an inner node, the classification uncertainty of attribute A is defined with the class label $\mathrm{C} = \{C_1, \dots, C_L\}$ as follows:

$$C_u(A, E) = \sum\nolimits_{1}^{n} \omega(E_i) C_u(E_i) \quad (10)$$

where E_i is the intersection of i-th linguistic value L_i of attribute A and path P from the root to the inner node. When $P = \emptyset, E = L_i$. $\omega(E_i)$ is a weight that represents the relative size of the subset E_i within P.

$$\omega(E_i) = M(E_i) / \sum_{i=1}^{n} M(P) \quad (11)$$

C. Stopping Criteria

A tree is learned by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner and is called recursive partitioning. Recursive partitioning is considered complete when a stopping criteria is reached.

From Definition 4, $\pi(C_j|L_i\ or\ E_i)$ represents the degree of truth for the classification rule "IF L_i or E_i THEN C_j". Therefore, we set the truth level threshold β as a stopping criteria to control the growth of the tree. For example, if the truth level threshold is set to $\beta = 0.7$ and the branch "Singer" ∩ the branch "[20-30]" results in E, the truth level of E for class "Coffee" is $0.83 > \beta$, so the branch "Singer" terminates as a leaf and "Coffee" is selected as its label. A lower β may lead to a smaller tree but with reduced classification accuracy. A higher β may lead to a larger tree with higher classification accuracy. However, when β increases to certain point, no further gains in accuracy will be derived. The settings for α and β depend on the particular situation.

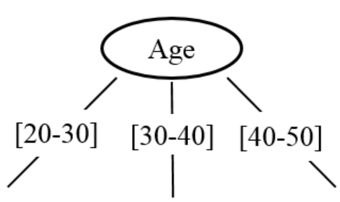

Figure 1. an attribute is a root node.

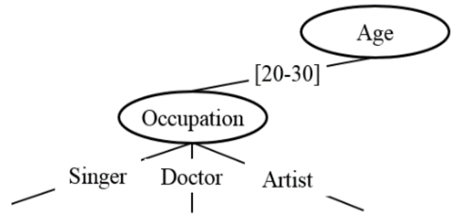

Figure 2. an attribute is a inner node.

D. Inducing the Fuzzy Decision Tree

This section presents the induction process of the fuzzy decision tree. The induction process for constructing decision trees take a top-down approach, by choosing the attributes with minimum classification uncertainty at each step, and set a truth level threshold β as stopping criteria. Hence, the induction process consists of the following steps:

Step 1: Measure the classification uncertainty associated with each attribute and select the attribute with the smallest classification uncertainty as the root decision node.

Step 2: Delete all empty branches of the decision node. For each non-empty branch of the decision node, calculate the truth level of all object classifications within the branch for each class. If the truth level of the classification into one class is above the specified threshold β, terminate the branch as a leaf. Otherwise, investigate whether an additional attribute would further partition the branch and select the attribute with smallest classification ambiguity as a new decision node from the branch. If not, terminate this branch as a leaf. At a leaf, label all objects as the class with the highest truth level.

Step 3: Repeat Step 2 for all newly generated decision nodes until no further growth is possible. This completes the decision tree.

Finally, our algorithm generates a fuzzy decision tree. Figure 3 gives an example of a branch of the fuzzy decision tree.

E. Fuzzy decision tree classification

Each branch from root to leaf can be converted into a fuzzy rule with a classification membership. Using the datasets in Table 1 as an example, a branch on the fuzzy decision tree is shown in Figure 3, the converted fuzzy rules are shown in Figure 4, and the values in brackets indicate classification membership. We use these fuzzy rules to classify new objects.

In fuzzy decision tree, many fuzzy rules can be applied at the same time, and an object may be classified into different class labels with different memberships. Therefore, we firstly determine what fuzzy rules are applied according to the object's attributes. When two or more rules are applied and classify the object into only one class, the class label with the highest membership is selected. When two or more rules are applied and classify the object into the different classes with different memberships, the class label, which has maximum classification membership, is taken as the object's class label. When two or more rules are applied and classify the object into the different classes with same memberships, the class label, which belongs to a leaf node at higher levels of the tree, is taken as the object's class label. For example, one object applies four fuzzy rules (Figure 4) at the same time, its class label is "Coffee".

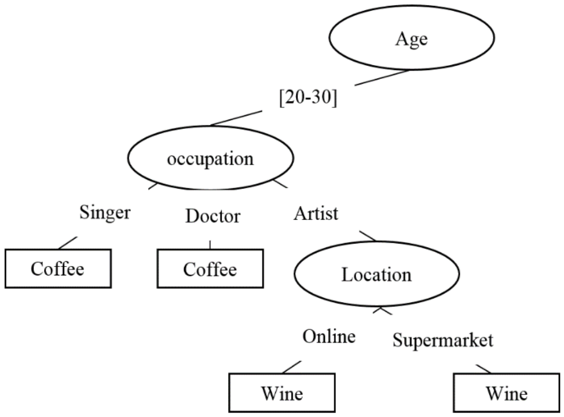

Figure 3. A branch of the fuzzy decision tree.

Rule 1: *IF Age IS [20-30] AND Occupation IS Singer THEN Coffee (0.83)*
Rule 2: *IF Age IS [20-30] AND Occupation IS Doctor THEN Coffee (0.73)*
Rule 3*: IF Age IS [20-30] AND Occupation IS Artist AND Location IS Online THEN Wine (0.83)*
Rule 4: *IF Age IS [20-30] AND Occupation IS Artist AND Location IS Supermarket THEN Wine (0.63)*

Figure 4. Fuzzy rules.

VI. EXPERIMENTS

In this section, we present the experimental results for our proposed fuzzy decision tree (FDT) algorithm. In a series of experiments, we compare our FDT algorithm to other existing algorithms by using multiple datasets, and we evaluate FDT's performance when the training data has uncertain classes. Our algorithms were implemented in Java, and all the evaluations were run on a Windows machine with an Intel 2.66 GHz Pentium(R) Dual-Core processor and 4 GB of main memory.

A. DataSets

Due to there is no real uncertain datasets available, the datasets we used in the experiments are synthesized from real datasets. All source datasets were taken from the UCI Machine Learning Repository [26] and Table II shows the type of dataset and the number of training and testing instances. In this paper, we adopt one of the most common synthetic method [5] to add uncertainty information.

TABLE II. DATASETS

Datasets	*Type*	*#Train*	*#Test*	*#Classes*
Iris	numerical	150	10-fold	3
Glass	numerical	214	10-fold	7
Segment	numerical	2310	10-fold	7
Voting	categorical	435	10-fold	2
Mushroom	categorical	2124	10-fold	2
Audiology	categorical	226	10-fold	24
Adult	mixed	2645	10-fold	2
Bridges	mixed	108	10-fold	7
Teaching	mixed	151	10-fold	3

From Table II, nine real-world datasets are used to evaluate our algorithm performance. Because some of the algorithms only deal with one type of data, we divided the datasets into three categories: categorical, numerical, and mixed. Mixed type data contains both categorical and numerical type data. Furthermore, we used 10-fold cross-validation to measure accuracy.

B. Performance Evaluation in certain class label

This section compares the FDT algorithm to the DTU [5], UDT [10], UBayes [8], USVM [6], and UELM [7] algorithms on above datasets. Three points are worth noting: (1) Most of the above algorithms use a Gaussian distribution as the uncertain model for UNAs, so we only used Gaussian distribution in our experiments. (2) The uncertainty of data's attribute is set to 20%. (3) Some parameters will affect the experimental results, but different values could be used with different datasets. Due to space limitations, we are not able to show the values of these parameters on different datasets, so 9/10 of data is used for training and 1/10 for testing. The whole procedure is repeated 10 times, and the overall accuracy rate is counted as the average of accuracy rates on each partition.

TABLE III. COMPARISON OF RESULTS (ONE TYPE)

Method	Numerical			Categorical		
	Iris	*Glass*	*Segment*	*Voting*	*Mushroom*	*Audiology*
DTU	92.36%	69.70%	90.80%	90.50%	89.48%	80.00%
UDT	96.00%	70.79%	92.23%	N/A		
UBayes	95.40%	47.15%	79.92%	N/A		
USVM	91.38%	63.25%	N/A	N/A		
UELM	92.04%	65.77%	N/A	N/A		
FID3	96.33%	72.04%	93.79%	91.39%	91.05%	81.98%

Table III shows that FDT was able to build more accurate decision trees than the other algorithms when ω was equal to 10%. Comparing the six row with other rows of Table III, the difference in accuracy is remarkable. Such as, the accuracy increased from 70.79% to 72.04% on the "Glass" dataset.

Due to UNAs and UNCs can both be represented in same terms by our discretization method, we use three real-world datasets (see Table II) to evaluate our FDT algorithm. The algorithms were evaluated in terms of accuracy and running time. Table IV shows FDT's performance for an uncertainty range of 0 to 20% and whether the probability distribution function is uniform or Gaussian.

TABLE IV. COMPARISON OF RESULTS (MIXED TYPE)

Datasets	***Gaussian distribution***			***Uniform distribution***		
	ω = 0%	***ω = 10%***	***ω = 20%***	***ω = 0%***	***ω = 10%***	***ω = 20%***
Adult	84.31%	82.71%	76.20%	83.92%	79.92%	75.17%
Bridges	67.47%	66.01%	61.67%	66.57%	65.39%	60.16%
Teaching	82.26%	81.02%	78.23%	81.87%	79.65%	77.11%

Overall, the accuracy of the FDT classifier remained relatively stable when dealing mixed types of data. Even when the extent of the uncertainty reached 20%, the accuracy is still quite comparable to that of precise data. These results demonstrate that FDT is quite robust against data uncertainty. We also observed similar trends for both uniform and Gaussian distributions of uncertain data. However, a classifier with a Gaussian PDF get better performance than with a uniform *PDF*.

C. Performance Evaluation in uncertain class label

To evaluate the performance of our FDT algorithm by considering class label uncertainty, we conduct an evaluation on all nine datasets as listed in Tables II in terms of accuracy and running time. The definition of accuracy was taken from [11], and we also use synthetic method [5] to add uncertainty information for class labels. Such as if 10% uncertainty is introduced, the attribute has a 90% probability of taking the original value and a 10% probability of taking one of the other values. Suppose in the original accurate dataset $A_{ij} = v_1$, then we will assign p_{j1} = 90%, and assign one p_{jl} ($2 \leq l \leq k$) = 10%. In these experiments, the uncertainty of the attribute data was set to 10%, and in order to show better results, we used Gaussian distribution as the uncertain model for the UNAs. The running time results for the six datasets is shown in Table V. Figure 4 shows the results for average accuracy.

TABLE V. RUNNING TIME

Datasets	**Running time (s)**		
	Training+Classificaiton		
	ω = 0%	***ω = 10%***	***ω = 20%***
Iris	0.068	0.269	0.568
Glass	0.249	0.452	0.750
Segment	1.217	1.415	1.720
Voting	0.084	0.287	0.589
Mushroom	1.093	1.303	1.694
Audiology	0.101	0.305	0.621
Adult	1.247	1.443	1.746
Bridges	0.087	0.285	0.589
Teaching	0.073	0.279	0.575

Tables V show that it generally takes longer to construct a classifier as the uncertainty in data increases. The reason is more candidate splitting points are available and require more comparisons for uncertain data.

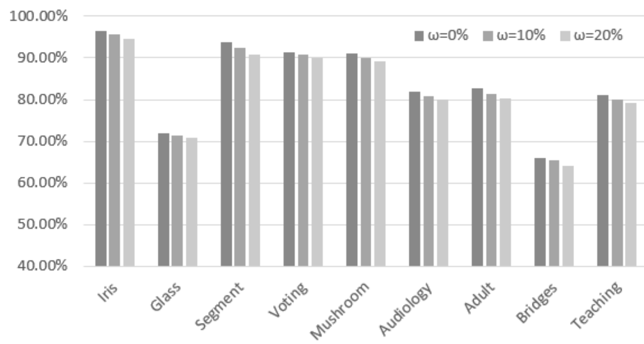

Figure 5. Example of a ONE-COLUMN figure caption.

As shown in Figure 1, when the extent of class uncertainty increases, the classifier accuracy declines slowly. For most datasets, the performance decrement is within 3%, even when class uncertainty reaches 30%. The "Bridges" identification dataset shows the worst performance with the classification accuracy over 66.01% on certain classes, but reducing to around 65.39% when the class uncertainty was 10%, and to 64.07% when the class uncertainty was 20%.

VII. CONCLUSION

In this paper, we address a special data classification problem called uncertain data classification problem. In this problem, both the attributes and the class label of a data are uncertain. To solve the problem, a fuzzy decision tree (FDT) algorithm is proposed in this paper. The algorithm use fuzzy set to represents data uncertainty and selected attributes with minimum classification uncertainty at each step. Our FDT algorithm has three advantages. Firstly, it can classify uncertain data that traditional data classification algorithms tend not to be able to deal with well. Secondly, it considers the uncertainty of class label, which is ignored in many existing uncertain classification algorithms. Finally, our proposed discretization method is able to convert interval data into a probability vector P, so that our algorithm can omit the difference between UNA and UCA. Extensive experiments on real-world datasets show that our FDT algorithm can produce classifiers with higher accuracy than some existing uncertain classification algorithms, regardless of whether an object's class label are uncertain.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Database mining: A performance perspective," IEEE transactions on knowledge and data engineering, vol. 5, Dec. 1993, pp. 914-925, doi:10.1109/69.250074.

[2] N. Cao, S. Choi, E. Masazade, and PK. Varshney, "Sensor selection for target tracking in wireless sensor networks with uncertainty," IEEE Transactions on Signal Processing, vol. 64, Oct. 2016, pp. 5191-5204, doi:10.1109/TSP.2016.2595500.

[3] RO. Chavez-Garcia and O. Aycard, "Multiple sensor fusion and classification for moving object detection and tracking," IEEE Transactions on Intelligent Transportation Systems, vol. 17, Feb. 2016, pp. 525-534, doi:10.1109/TITS.2015.2479925.

[4] J. Beekhuizen, GB. Heuvelink, A. Huss, A. Bürgi, H. Kromhout, and R. Vermeulen, "Impact of input data uncertainty on environmental exposure assessment models: A case study for electromagnetic field modelling from mobile phone base stations," Environmental research vol. 135, Nov. 2014, pp. 148-155, doi:10.1016/j.envres.2014.05.038.

[5] B. Qin, Y. Xia, and F. Li, "DTU: a decision tree for uncertain data," Advances in Knowledge Discovery and Data Mining, Springer Press, 2009, pp. 4-15, doi:10.1007/978-3-642-01307-2_4.

[6] J. Bi. and T. Zhang, "Support vector classification with input data uncertainty," International Conference on Neural Information Processing Systems, MIT Press, 2005, pp. 161-168.

[7] Y. Sun, Y. Yuan, and G. Wang, "Extreme learning machine for classification over uncertain data," Neurocomputing, vol. 128, Mar. 2014, pp. 500-506, doi:10.1016/j.neucom.2013.08.011.

[8] B. Qin, Y. Xia, S. Wang, and X. Du, "A novel Bayesian classification for uncertain data," Knowledge-Based Systems vol. 24, Dec 2011, pp. 1151-1158, doi:10.1016/j.knosys.2011.04.011.

[9] B. Qin, Y. Xia, S. Prabhakar, and Y. Tu, "A rule-based classification algorithm for uncertain data." International Conference on Data Engineering, IEEE Press, Mar 2009, pp. 1633-1640, doi:10.1109/ ICDE.2009.164

[10] S. Tsang, B. Kao, KY. Yip, WS. Ho, and SD. Lee, "Decision trees for uncertain data." IEEE transactions on knowledge and data engineering, vol. 23, Jan. 2011, pp. 64-78, doi:10.1109/TKDE.2009.175

[11] C. Jin, F. Li, and Y. Li, "A generalized fuzzy ID3 algorithm using generalized information entropy," Knowledge-Based Systems vol. 64 Jul 2014, pp. 13-21, doi:10.1016/j.knosys.2014.03.014.

[12] M. Pratama, J. Lu, G. Zhang, "Evolving type-2 fuzzy classifier," IEEE Transactions on Fuzzy Systems, vol. 24, Jun 2016, pp. 574-589, doi: 10.1109/TFUZZ.2015.2463732.

[13] X. Wang, X. Liu, W. Pedrycz, and L. Zhang, "Fuzzy rule based decision trees," Pattern Recognition, vol. 48, Jan 2015, pp. 50-59, doi:10.1016/j.patcog.2014.08.001.

[14] Y. Yuan and MJ. Shaw, "Induction of fuzzy decision trees," Fuzzy Sets and systems, vol.69, Jan 1995, pp. 125-139, doi: 10.1016/0165-0114(94)00229-Z.

[15] C. Olaru, and L. Wehenkel, "A complete fuzzy decision tree technique," Fuzzy sets and systems, vol. 138, Sep. 2003, pp. 221-254, doi: 10.1016/S0165-0114(03)00089-7.

[16] JR. Quinlan, "Generating Production Rules From Decision Trees," International Joint Conference on Artificial Intelligence, vol. 1, 1987.

[17] LA. Kurgan and KJ. Cios, "CAIM discretization algorithm," IEEE transactions on Knowledge and Data Engineering, vol. 16, Feb. 2004 pp. 145-153, doi: 10.1109/TKDE.2004.1269594.

[18] LY. Wen, F. Min F, SY. Wang, "A two-stage discretization algorithm based on information entropy," Applied Intelligence, May. 2017, pp. 1-17, doi: 10.1007/s10489-017-0941-0.

[19] L. Zou, D. Yan, and HR. Karimi, "An algorithm for discretization of real value attributes based on interval similarity," Journal of Applied Mathematics, Apr. 2013, doi:10.1155/2013/350123.

[20] H. Zheng, J. He, Y. Zhang, and Y. Shi, "A fuzzy decision tree approach based on data distribution construction." In Proceedings of the Australasian Computer Science Week Multiconference, Jan 2017, p. 5, doi:10.1145/3014812.3014817.

[21] X. Wang, B. Chen, G. Qian, and F. Ye, "On the optimization of fuzzy decision trees," Fuzzy Sets and Systems vol. 112, May 2000, pp. 117-125, doi:10.1016/S0165-0114(97)00386-2.

[22] Yuan, and GJ. Klir, "Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A. Zadeh," Archive for Mathematical Logic, vol. 32, Sep 1996, pp. 1-32.

[23] G. Deschrijver and EE. Kerre, "On the relationship between some extensions of fuzzy set theory," Fuzzy sets and systems, vol. 133, Jan 2003, pp. 227-235, doi:10.1016/S0165-0114(02)00127-6.

[24] S. Das, S. Ghosh, S. Kar, and T. Pal, "An Algorithmic Approach for Predicting Unknown Information in Incomplete Fuzzy Soft Set," Arabian Journal for Science and Engineering, May 2017, pp. 1-9, doi: 10.1007/s13369-017-2591-2.

[25] LA. Zadeh, "Fuzzy sets as a basis for a theory of possibility," Fuzzy sets and systems, vol. 100, Feb 1999, pp. 9-34, doi:10.1016/S0165-0114(99)80004-9.

[26] C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998. [Online]. Available: ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩.