DTU: A Decision Tree for Uncertain Data

Biao Qin[1], Yuni Xia[1], and Fang Li[2]

[1] Department of Computer and Information Science, Indiana University - Purdue University Indianapolis, USA
{biaoqin,yxia}@cs.iupui.edu
[2] Department of Mathematics, Indiana University - Purdue University Indianapolis, USA
lfang@iupui.edu

Abstract. Decision Tree is a widely used data classification technique. This paper proposes a decision tree based classification method on uncertain data. Data uncertainty is common in emerging applications, such as sensor networks, moving object databases, medical and biological bases. Data uncertainty can be caused by various factors including measurements precision limitation, outdated sources, sensor errors, network latency and transmission problems. In this paper, we enhance the traditional decision tree algorithms and extend measures, including entropy and information gain, considering the uncertain data interval and probability distribution function. Our algorithm can handle both certain and uncertain datasets. The experiments demonstrate the utility and robustness of the proposed algorithm as well as its satisfactory prediction accuracy.

1 Introduction

Decision trees is a simple yet widely used method for classification and predictive modeling. A decision tree partitions data into smaller segments called terminal nodes. Each terminal node is assigned a class label. The non-terminal nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. The partitioning process terminates when the subsets cannot be partitioned any further using predefined criteria. Decision trees are used in many domains. For example, in database marketing, decision trees can be used to segment groups of customers and develop customer profiles to help marketers produce targeted promotions that achieve higher response rates.

This paper studies decision tree based classification methods for uncertain data. In many applications, data contains inherent uncertainty. A number of factors contribute to the uncertainty, such as the random nature of the physical data generation and collection process, measurement and decision errors, unreliable data transmission and data staling. For example, there are massive amounts of uncertain data in sensor networks, such as temperature, humidity, and pressure. Uncertainty can also arise in categorical data. For instance, a tumor is typically classified as benign or malignant in cancer diagnosis and treatment.

T. Theeramunkong et al. (Eds.): PAKDD 2009, LNAI 5476, pp. 4–15, 2009.

In practice, it is often very difficult to accurately classify a tumor due to the experiment precision limitation. The lab results inevitably give false positives or false negatives some of the time. Therefore, doctors may often decide tumors to be benign or malignant with certain probability or confidence. [24]

Since data uncertainty is ubiquitous, it is important to develop classification models for uncertain data. In this paper, We focus on the decision tree based classification approach. We choose the decision tree because of its numerous positive features. Decision tree is simple to understand and interpret. It requires little data preparation, while some other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Decision tree can handle both numerical and categorical data, while many other techniques are usually specialized in analyzing datasets that have only one type of variable. Decision tree uses a white box model. If a given situation is observable in a model the explanation for the condition is easily explained by Boolean logic. Besides, it is possible to validate a decision tree model using statistical tests. Decision tree is also robust and scalable. It performs well with large data in a short period of time.

In this paper, we propose a new decision tree for classifying and predicting both certain and uncertain data (DTU). The main contributions of this paper are:

1. We integrate the uncertainty data model into the design of the decision tree.
2. We develop the DTU based on the widely used C4.5 classification tree so that it can handle both numerical and categorical data with uncertainty.
3. We prove through experiments that DTU has satisfactory performance even when the training data is highly uncertain.

This paper is organized as follows. In the next section, we will discuss related work. Section 3 describes the uncertaint data model. Section 4 shows the measures for identifying the best split for uncertain data. Section 5 illustrates the DTU algorithms in detail. The experimental results are shown in Section 6 and Section 7 concludes the paper.

2 Related Work

Classification is a well-studied area in data mining. Many classification algorithms have been proposed in the literature, such as decision tree classifiers [17], Bayesian classifiers [14], support vector machines (SVM) [20], artificial neural networks [3] and ensemble methods [9]. In spite of the numerous classification algorithms, building classification based on uncertain data has remained a great challenge. There are early work performed on developing decision trees when data contains missing or noisy values [11,15,18]. Various strategies have been developed to predict or fill missing attribute values. However, the problem studied in this paper is different from before - instead of assuming part of the data has missing or noisy values, we allow the whole dataset to be uncertain, and the uncertainty is not shown as missing or erroneous values but represented as uncertain intervals and probability distribution functions. There are also some previous work performed on classifying uncertain data in various applications [4,10,12]. All of the above methods try to solve specific classification tasks instead of developing a general algorithm for classifying uncertain data. And Qin et al. [24] propose a rule-based classification algorithm for uncertain data.

Recently, more research has been conducted in uncertain data mining. Most of them focus on clustering uncertain data [8,13,16]. The key idea is that when computing the distance between two uncertain objects, the probability distributions of objects are used to compute the expected distance. Xia et al. [22] introduce a new conceptual clustering algorithm for uncertain categorical data. Aggarwal [2] proposes density based transforms for uncertain data mining. There is also some research on identifying frequent itemsets and association mining [7,23] from uncertain datasets. The support of itemsets and confidence of association rules are integrated with the existential probability of transactions and items. Burdicks [5] discuss OLAP computation on uncertain data. None of them address the issue of developing a general classification and prediction algorithm for uncertain data.

3 Data Uncertainty

In this section, we will discuss the uncertainty model for both numerical and categorical attributes. Here we focus on the attributes uncertainty and assume the class type is certain.

When the value of a numerical attribute is uncertain, the attribute is called an uncertain numerical attribute (UNA), denoted by $A_i^{u_n}$. Further, we use $A_{ij}^{u_n}$ to denote the jth instance of $A_i^{u_n}$. The concept of UNA has been introduced in [6]. The value of $A_i^{u_n}$ is represented as a range or interval and the probability distribution function (PDF) over this range. Note that $A_i^{u_n}$ is treated as a continuous random variable. The PDF $f(x)$ can be related to an attribute if all instances have the same distribution, or related to each instance if each instance has a different distribution.

An uncertain interval instance of $A_i^{u_n}$, denoted by $A_{ij}^{u_n}.U$, is an interval $[A_{ij}^{u_n}.l, A_{ij}^{u_n}.r]$ where $A_{ij}^{u_n}.l, A_{ij}^{u_n}.r \in R$, $A_{ij}^{u_n}.r \geq A_{ij}^{u_n}.l$. The uncertain PDF of $A_{ij}^{u_n}$, denoted by $A_{ij}^{u_n}.f(x)$, is a probability distribution function of $A_{ij}^{u_n}$, such that $\int_{A_{ij}^{u_n}.l}^{A_{ij}^{u_n}.r} A_{ij}^{u_n}.f(x)dx = 1$ and $\int_{A_{ij}^{u_n}.l}^{A_{ij}^{u_n}.r} A_{ij}^{u_n}.f(x)dx = 0$ if $x \notin A_{ij}^{u_n}.U$.

A dataset can also have categorical attributes that are allowed to take on uncertain values. We call such attributes uncertain categorical attributes(UCA), denoted by $A_i^{u_c}$. Further, we use $A_{ij}^{u_c}$ to denote the attribute value of the jth instance of $A_i^{u_c}$. The notion of UCA was proposed in [19].

$A_{ij}^{u_c}$ takes values from the categorical domain Dom with cardinality $|Dom| = n$. For a certain dataset, the value of an attribute A is a single value d_k in Dom, $Pr(A = d_k) = 1$. In the case of an uncertain dataset, we record the information by a probability distribution over Dom instead of a single value. Given a categorical domain $Dom = \{d_1, \ldots, d_n\}$, an uncertain categorical attribute (UCA)

A^{u_c} is characterized by probability distribution over Dom. It can be represented by the probability vector $P = \{p_1, \ldots, p_n\}$ such that $P(A_{ij}^{u_c} = v_k) = p_{jk}$ and $\sum_{k=1}^{n} p_{jk} = 1 (1 \leq k \leq n)$.

4 Attribute Test Condition: Identifying the Best Split

The key issue of a decision tree induction algorithm is to decide the way records be split. Each step of the tree-grow process needs to select an attribute test condition to divide the records into smaller subsets. Widely used splitting measures such as information entropy and the Gini index are not applicable to uncertain data. In this section, we will define splitting measures for both uncertain numerical data and uncertain categorical data.

4.1 Uncertain Numerical Attributes

As described earlier, the value of an uncertain numerical attribute is an interval with associated PDF. Table 1 shows an example of UNA. The data in this table are used to predict whether borrowers will default on loan payments. Among all the attributes, the Annual Income is an UNA, whose precise value is not available. We only know the range of the Annual Income of each person and the PDF $f(x)$ over that range. The probability distribution function of the UNA attribute Annual Income is assumed to be uniform distribution.

Each uncertain numerical value has a maximal value and a minimal value, which we call critical points. For each UNA, we can order all critical points of an uncertain numerical attribute in an ascending sort with duplicate elimination. Then the UNA can be partitioned. One partition may overlap with the UNA of many instances. When an instance with UNA overlaps with a partition [a, b),

Table 1. Uncertain Numerical Data

ID	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	110-120	No
2	No	Married	100-120	No
3	No	Single	60-85	No
4	Yes	Married	110-145	No
5	No	Divorced	110-120	Yes
6	No	Married	50-80	No
7	Yes	Divorced	170-250	No
8	No	Single	85-100	Yes
9	No	Married	80-100	No
10	No	Single	120-145	Yes
11	No	Divorced	105-125	Yes
12	No	Divorced	80-95	No

the probability its UNA actually falls in that partition is $\int_a^b f(x)dx$. Based on the probability of each individual instance falling in a partition [a, b), we can compute the probabilistic number of instances falling in that partition, which we call probabilistic cardinality.

The probabilistic cardinality of the dataset over a partition Pa = [a, b) is the sum of the probabilities of each instance whose corresponding UNA falls in [a, b). That is, $PC(Pa) = \sum_{j=1}^{n} P(A_{ij}^{u_n} \in [a,b)) = \sum_{j=1}^{n} \int_a^b A_{ij}^{u_n}.f(x)dx$. The probabilistic cardinality for class C_j of the dataset over a partition Pa=[a, b) is the sum of the probability of each instance T_j in C_j whose corresponding UNA falls in [a, b). That is, $PC(Pa, C) = \sum_{j=1}^{n} P(A_{ij}^{u_n} \in [a,b) \wedge C_{T_j} = C_j)$, where $C_{T_j} = C_j)$ denotes the class label of instance T_j.

Refer to the dataset in Table 1, the probabilistic cardinality for the partition [110, 120) on the Annual Income is the sum of the probabilities of instances that have Annual Income falling in [110, 120). Suppose the annual income for each instance is uniformly distributed over its uncertain interval; instances 1, 2, 4, 5 and 11 have overlap with [110,120), and the probability for instance 1 with annual income in [110, 120) is $P(I1 \in [110, 120)) = (120-110)/(120-110) = 1$. Similarly, $P(I2 \in [110, 120)) = 0.5$, $P(I4 \in [110, 120)) = 0.29$, $P(I5 \in [110, 120)) = 1$, and $P(I11 \in [110, 120)) = 0.5$; therefore, the probabilistic cardinality for this dataset over partition [110, 120) is 3.29. The probabilistic cardinality for class DefaultBorrower = NO over the partition [110, 120) on the Annual Income is the sum of the probabilities of instances who are not DefaultBorrowers with Annual Income falling in [110, 120). Among instances 1, 2, 4, 5 and 11 who have overlap with [110,120), only instances 1, 2 and 4 are in class NO; therefore, the probabilistic cardinality for DefaultBorrower = NO over partition [110, 120) is 1.79. Similarly, the probabilistic cardinality for DefaultBorrower = Yes over partition [110, 120) is 1.5.

With the two previous definitions, we can now define the probabilistic entropy for uncertain data as follows:

Definition 1. *The* Probabilistic Entropy *for a dataset D is* $ProbInfo(D) = -\sum_{i=1}^{m} \frac{PC(D,i)}{PC(D)} \times log_2(\frac{PC(D,i)}{PC(D)})$.

Suppose attribute A is selected as the split attributecan, and it partitions the dataset D into k subsets, $\{D_1, D_2, \ldots, D_k\}$. Then the probabilistic entropy, or expected information based on the partitioning is given by $ProbInfo_A(D) = \sum_{j=1}^{k} \frac{PC(D_j)}{PC(D)} \times ProbInfo(D_j)$. The term $PC(D_j)$ acts as the weight of the jth partition. The smaller the entropy value, the greater the purity of the subset partitions. The encoding information that would be gained by branching on A is $ProbGain(A) = ProbInfo(D) - ProbInfo_A(D)$.

Probabilistic Entropy also tends to favor attributes that have a large number of distinct values. The information gained by a test is maximal when there is one case in each subset D_j. To overcome this problem, the splitting criterion should be modified to take into account the number of outcomes produced by the attribute test condition. This criterion is defined as $ProbGain\_ratio(A) = \frac{ProbGain(A)}{ProbSplitInfo_A(D)}$. Here, $ProbSplitInfo_A(D) = -\sum_{j=1}^{k} \frac{PC(D_j)}{PC(D)} \times log_2(\frac{PC(D_j)}{PC(D)})$

and k is the total number of splits. If an attribute produces a large number of splits, its split information will also be large, which in turn reduces its gain ratio.

4.2 Uncertain Categorical Data

An uncertain discrete attribute (UCA) $A_i^{u_c}$ is characterized by probability distribution over Dom. As mentioned earlier, it can be represented by the probability vector $\{p_1, \ldots, p_n\}$ such that $P(A_{ij}^{u_c} = d_j) = p_j (1 \leq j \leq n)$.

Table 2. Uncertain Categorical Data

ID	Make	Date	Problem	Location	Class
1	Explorer	4/5/08	(Brake:0.5; Tire:0.5)	CA	0
2	Camry	8/3/02	(Trans:0.2; Tire:0.8)	IN	1
3	Civic	9/12/99	(Exhaust:0.4; Brake:0.6)	TX	0
4	Pontiac	4/2/01	(Tire:1.0)	IL	1
5	Caravan	1/23/04	(Trans:0.3; Brake:0.7)	NY	1

Table 2 shows an example of UCA [19]. This dataset records vehicle problem information. The problem can be caused by the brake, tire, transmission or other parts. It is derived from the text field in the given tuple using a text classifier/miner. As text miner result tend to be uncertain, the Problem field is a UCA.

Similar to uncertain numerical data, the probabilistic cardinality of the dataset over d_j is the sum of the probabilities of each instance whose corresponding UCA equals to d_j. That is, $PC(d_j) = \sum_{j=1}^{n} P(A_{ij}^{u_c} = d_j)$. The probabilistic cardinality for class C of the dataset over d_j is the sum of the probabilities of each instance in C_j whose corresponding UCA equals to d_j. That is, $PC(d_j, C) = \sum_{j=1} P(A_{ij}^{u_c} = d_j \wedge C_j = C)$.

Refer to the dataset in Table 2, the probabilistic cardinality over Problem = Brake is the sum of the probabilities of each instance whose Problem attribute is Brake, which is 1.8. The probabilistic cardinality for class 0 over "Problem = Brake" is the overall probabilities of instances in class 0 whose Problem attribute is Brake, which is 1.1. Based on the probabilistic cardinality for each class C, we can then compute the probabilistic information entropy and probabilistic information gain ratio if the data is split on the categorical attribute "Problem", following the same process as for uncertain numerical data. If it has the highest probabilistic information gain, then "Problem" will be chosen as the next splitting attribute.

5 Algorithms for DTU

5.1 Decision Tree Induction Algorithm

The algorithm is shown in Algorithm 1. The basic strategy is as follows:

Algorithm 1. DTU Induction

input: the training dataset D; the set of candidate attributes att-list
output: An uncertain decision tree
begin
1: create a node N;
2: **if** (D are all of the same class, C) **then**
3: return N as a leaf node labeled with the class C;
4: **else if** (attribute-list is empty) **then**
5: return N as a leaf node labeled with the highest weight class in D;
6: **end if**;
7: select a test-attribute with the highest probabilistic information gain ratio to label node N;
8: **if** (test-attribute is numeric or uncertain numeric) **then**
9: binary split the data from the selected position y;
10: **for** (each instance R_j) **do**
11: **if** (test-attribute $\leq y$) **then**
12: put it into D_l with weight $R_j.w$;
13: **else if** (test-attribute $> y$) **then**
14: put it into D_r with weight $R_j.w$;
15: **else**
16: put it into D_l with weight $R_j.w * \int_{x_1}^{y} f(x)dx$;
17: put it into D_r with weight $R_j.w * \int_{y}^{x_2} f(x)dx$;
18: **end if**;
19: **end for**;
20: **else**
21: **for** (each value $a_i (i = 1, \ldots, n)$ of the attribute) **do**
22: grow a branch D_i for it;
23: **end for**;
24: **for** (each instance R_j) **do**
25: **if** (test-attribute is uncertain) **then**
26: put it into D_i with $R_j.a_i.w * R_j.w$ weight;
27: **else**
28: put it into a certain D_i with weight $R_j.w$;
29: **end if**
30: **end for**;
31: **end if**;
32: **for** each D_i **do**
33: attach the node returned by DTU(D_i, att-list);
34: **end for**;
end

1. The tree starts as a single node representing the training samples (step 1).

2. If the samples are all of the same class; then the node becomes a leaf and is labeled with that class (steps 2 and 3).

3. Otherwise, the algorithm uses a probabilistic entropy-based measure, known as the probabilistic information gain ratio, as the criteria for selecting the attribute that will best separate the samples into an individual class (step 7). This attribute becomes the "test" attribute at the node.

4. If the test attribute is numerical or uncertain numerical, we split for the data at the selected position y (steps 8 and 9).

5. A branch is created for test-attribute $\leq y$ or test-attribute $> y$ respectively. If an instance's test attribute value $[x_1, x_2]$ is less than or equal to y ($x_2 \leq y$), it is put into the left branch with the instance's weight $R_j.w$. If an instance's test attribute value $[x_1, x_2]$ is larger than y ($x_1 > y$), it is put into the right branch with the instance's weight $R_j.w$. If an attribute's value $[x_1, x_2]$ covers the split point y ($x_1 \leq y < x_2$), it is put into the left branch with weight $R_j.w * \int_{x_1}^{y} f(x)dx$ and the right branch with weight $R_j.w * \int_{y}^{x_2} f(x)dx$. Then the dataset is divided into D_l and D_r (steps 10-19).

6. If the test attribute is categorical or uncertain categorical, we split the data multiway (steps 21-30). A branch is created for each value of the test attribute, and the samples are partitioned accordingly. For each value a_i of the attribute, an instance is put into D_i with $R_j.w$ weight when the attribute is certain. If the attribute is uncertain, assume the probability of the attribute value a_i be $R_j.a_i.p$, then the instance is put into the branch a_i with the weight $R_j.a_i.p * R_j.w$.

7. The algorithm recursively applies the same process to generate a decision tree for the samples.

8. The recursive partitioning process stops only when either of the following conditions becomes true:

1) All samples for a given node belong to the same class (steps 2 and 3), or

2) There are no remaining attributes on which the samples may be further partitioned (step 4). In this case, the highest weight class is employed (step 5). This involves converting the given node into a leaf and labeling it with the class having the highest weight among samples. Alternatively, the class distribution of the node samples may be stored.

5.2 Prediction with DTU

Once a DTU is constructed, it can be used for predicting class types. The prediction process starts from the root node, the test condition is applied at each node in DTU, and the appropriate branch is followed based on the outcome of the test. When the test instance R is certain, the process is quite straightforward since the test result will lead to one single branch without ambiguity. When the test is on an uncertain attribute, the prediction algorithm proceeds as follows:

1. If the test condition is on a UNA attribute A and the splitting point is a, suppose $R.A$ is an interval $[x_1, x_2)$ with associated pdf $R.A.f(x)$:

If $a < x_1$, which means the minimal possible value of $R.A$ is larger than a, then $P(R.A > a) = R.w$; we know for sure $R.A > a$ and R follows the right branch;

If $a >= x_2$, which means the maximal possible value of $R.A$ is smaller than a, then $P(R.A < a) = R.w$, and it is certain that $R.A < a$ and R follows the left branch;

If ($x_1 < a < x_2$), then the probability $R.A < a$ is $P(R.A < a) = R.w * \int_{x_1}^{a} f(x)dx$ and the probability $R.A > a$ is $P(R.A > a) = R.w * \int_{a}^{x_2} f(x)dx$. R

should be in the left branch with probability $R.w * \int_{x_1}^{a} f(x)dx$ and in the right branch with probability $R.w * \int_{a}^{x_2} f(x)dx$.

2. If the test condition is on a UCA attribute A and $a_1, a_2, ...a_k$ are the values for the categorical attribute A, then suppose $R.A$ is an UCA, that is $R.A = \{p_1, p_2, ..., p_k\}$, with $p_i(i = 1, ..., k)$ as the probability of $R.A = a_i$. Then R should be in the ith branch with probability p_i.

For the leaf node of DTU, each class C_i has a probability $PL(C_i)$, which is the probability for an instance to be in class C_i if it falls in this leaf node. $PL(C_i)$ is computed as the fraction of the probabilistic cardinality of instances in class C_i in a leaf node over the total probabilistic cardinality of instances in that node. Assume path L from the root to a leaf node contains t tests, and the data are classified into one class c_i in the end, suppose $P(T_i)$ is the probability that an instance follow the path at the ith test, then the probability for an instance to be in class c_i taking that particular path L is $P_{c_i}^{L} = PL(c_i) * \prod_{i=1}^{t} P(T_i)$.

When predicting the class type for an instance T with uncertain attributes, it is possible that the process takes multiple paths. Suppose there are m paths taken in total, then the probability for T in class c_i is $P_{c_i} = \sum_{i=1}^{m} P_{c_i}^{i}$.

Finally, the instance will be predicted to be of class c_i which has the largest P_{c_i} among all $P_{c_i}, i = 1, ..., n$.

6 Experiments

In this section, we present the experimental results of the proposed decision tree algorithm DTU. We studied the prediction accuracy over multiple datasets.

Based on the J4.8/C4.5 implemented on Weka [21], we implemented the DTU as described in Section 5. The experiments are executed on a PC with an Intel Pentium IV 3.4GHz CPU and 2.0 GB main memory. A collection containing 10 real-world benchmark datasets were assembled from the UCI Repository [1]. We tried to cover the spectrum of properties such as size, attribute numbers and types, number of classes and class distributions. Among these 10 datasets, 5 of them, namely Iris, Sonar, Segment, Diabetes and Glass contain mainly numerical attributes. The remaining 5 datasets, namely Audiology, Bridges, Promoters, Mushroom and voting have mostly categorical attributes.

Due to a lack of real uncertain datasets, we introduce synthetic uncertainty into the datasets. To make numerical attributes uncertain, we convert each numerical value to an uncertain interval with uniform probability distribution function. The uncertain interval is randomly generated around the original value. These are uncertainties from random effects without any bias. If the uncertain interval is within 10% of the original data, we call the dataset with uncertainty 10% and denote it by U10. For example, when the original value is 20, then its U10 may be [18.4, 20.4). We make categorical attributes uncertain by converting them into probability vectors. For example, a categorical attribute A_i may have k possible values $v_j, 1 \leq j \leq k$. For an instance I_j, we convert its value A_{ij} into a probability vector $\mathbf{P} = (p_{j1}, p_{j2}, ..., p_{ji}, ..., p_{jk})$, while p_{jl} is the probability of $A_{ij}^{u_c}$ to be equal to v_l, that is, $P(A_{ij}^{u_c} = v_l) = p_{jl}$. For example, when we

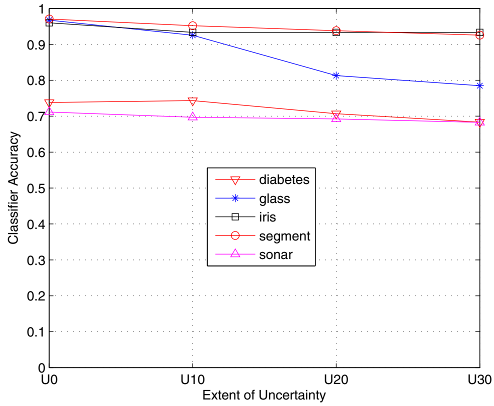

Fig. 1. DTU accuracy on uncertain numerical data sets

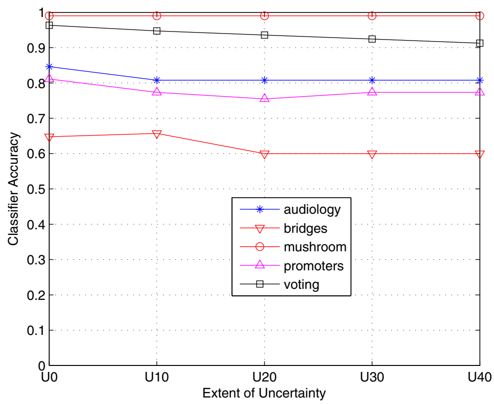

Fig. 2. DTU accuracy on uncertain categorical data sets

introduce 10% uncertainty, this attribute will take the original value with 90% probability, and 10% probability to take any of the other values. Suppose in the original accurate dataset $A_{ij} = v_1$, then we will assign $p_{j1} = 90\%$, and assign $p_{jl}(2 \le l \le k)$ to ensure $\sum_{i=2}^{k} p_{jl} = 10\%$. Similarly, we denote this dataset with 10% uncertainty in categorical data by U10. We use U0 to denote accurate or certain datasets.

As prediction accuracy is by far the most important measure of a classifier, we studied the prediction accuracy of DTU classifier first. Figure 1 shows the result for numerical datasets and Figure 2 shows the result for categorical datasets. In both experiments, we use ten-fold cross validation. Data is split into 10 approximately equal partitions; each one is used in turn for testing while the remainder is used for training, that is, 9/10 of data is used for training and 1/10 for test. The whole procedure is repeated 10 times, and the overall accuracy rate is counted as the average of accuracy rates on each partition. When DTU is applied on certain data, it works as a traditional C4.5 classifier.

For numerical data, the uncertainty varies between 0 to 30%. As shown in Figure 1, when the extent of uncertainty increases, the classifier accuracy declines slowly. For most datasets, the performance decrement are within 5%, even when data uncertainty reaches 30%. The worst performance decrement is for the glass identification dataset, the classifier has over 95% accuracy on certain data, reduces to around 92% when the uncertainty is 10%, to 81% when the uncertainty is 20%, and to 78% when the uncertainty reaches 30% .

The results for categorical datasets are similar, as shown in Figure 2. Overall speaking, the accuracy of DTU classifier remains relatively stable. The overall decrease in classifier accuracy is within 10% even when the uncertainty reaches 40%. Both experiments show DTU is quite robust against data uncertainty.

7 Conclusions

In this paper, we propose a new decision tree algorithm DTU for classifying and predicting uncertain data. We extend the measures used in tradition decision tree, such as information entropy and information gain, for handling data uncertainty. Our experiments demonstrate that DTU can process both uncertain numerical data and uncertain categorical data. It can achieve satisfactory classfication and prediction accuracy even when data is highly uncertain.

References

1. `http://archive.ics.uci.edu/ml/datasets.html`
2. Aggarwal, C.: On density based transforms for uncertain data mining. In: ICDE, pp. 866–875 (2007)
3. Andrews, R., Diederich, J., Tickle, A.: A survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge Based Systems 8(6), 373–389 (1995)
4. Bi, J., Zhang, T.: Support Vector Classification with Input Data Uncertainty. Advances in Neural Information Processing Systems 17, 161–168 (2004)
5. Burdick, D., Deshpande, M.P., Jayram, T.S., Ramakrishnan, R., Vaithyanathan, S.: OLAP over uncertain and imprecise data. The VLDB Journal 16(1), 123–144 (2007)
6. Cheng, R., Kalashnikov, D., Prabhakar, S.: Evaluating probabilistic queries over imprecise data. In: Proceedings of the ACM SIGMOD, pp. 551–562 (2003)
7. Chui, C., Kao, B., Hung, E.: Mining Frequent Itemsets from Uncertain Data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS, vol. 4426, pp. 47–58. Springer, Heidelberg (2007)
8. Cormode, G., McGregor, A.: Approximation algorithms for clustering uncertain data. In: PODS 2008, pp. 191–199 (2008)
9. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)

10. Gonzalez, E.V., Broitman, I.A.E., Vallejo, E.E., Taylor, C.E.: Targeting Input Data for Acoustic Bird Species Recognition Using Data Mining and HMMs. In: Proceedings of the ICDMW 2007, pp. 513–518 (2007)
11. Hawarah, L., Simonet, A., Simonet, M.: Dealing with Missing Values in a Probabilistic Decision Tree during Classification. In: The Second International Workshop on Mining Complex Data, pp. 325–329 (2006)
12. Jebari, C., Ounelli, H.: Genre categorization of web pages, In: Proceedings of the ICDMW 2007, pp. 455–464 (2007)
13. Kriegel, H., Pfeifle, M.: Density-Based Clustering of Uncertain Data. In: Proceedings of the KDD 2005, pp. 672–677 (2005)
14. Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. In: Proceedings of the tenth National Conference on artigicial intelligence, pp. 223–228 (1992)
15. Lobo, O., Numao, M.: Ordered estimation of missing values. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS, vol. 1574, pp. 499–503. Springer, Heidelberg (1999)
16. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient Clustering of Uncertain Data. In: Proceedings of ICDM 2006, pp. 436–445 (2006)
17. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufman Publishers, San Francisco (1993)
18. Quinlan, J.R.: Probabilistic decision trees. Machine Learning: an Artificial Intelligence Approach 3, 140–152 (1990)
19. Singh, S., Mayfield, C., Prabhakar, S., Shah, R., Hambrusch, S.: Indexing Categorical data with uncertainty. In: Proceedings of ICDE 2007, pp. 616–625 (2007)
20. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
21. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufman Publishers, San Francisco (2005)
22. Xia, Y., Xi, B.: Conceptual clustering categorical data with uncertainty. In: Proceedings of international conference on tools with artificial intelligence, pp. 329–336 (2007)
23. Yu, Z., Wong, H.: Mining Uncertain Data in Low-dimensional Subspace. In: Proceedings of ICPR 2006, pp. 748–751 (2006)
24. Qin, B., Xia, Y., Prbahakar, S., Tu, Y.: A Rule-based Classification Algorithm for Uncertain Data. In: The Workshop on Management and Mining Of Uncertain Data (MOUND) (2009)