ISSN:0975-9646

Distribution Based Analysis of Range-Bound Feature Values for Decision Tree Generation

Sudhakar R Lakkireddy[#] Sudheerbabu Pakanati [*] Rajani Devi. K[#]

[#]*Dept. of Computer Science & Engineering*
Nalanda Institute of Technology
Sattenapalli-522403,A.P.,India

[*]*Synova Innovative Technologies Pvt. Ltd.*
Bangalore- 560029, India

Abstract **- Classification of precise or point valued data is the regular practice in traditional approaches. If the data varies from point valued to a range bound values makes the classification complicated. In this scenario of uncertain data usual practice is to match the range bound values to single values by considering their mean or average. This approach however sacrifices the accuracy of the classifying the tuples to their associated class labels. The classical decision tree technique can be extended to handle the range bound values by considering the probability density function (pdf) over complete information of a feature or attribute. In this paper we propose distribution-based approach for classifying the uncertain data over decision tree. This technique is more accurate than traditional approach and incurs more computational cost. We produce empirical results supporting distribution based approach over traditional approach.**

Keywords **– uncertain data, decision tree, distribution-based.**

1. INTRODUCTION

Any database with external environments interaction faces uncertainty as a major problem. In particular, database systems that model and capture the state of any set up or real world entities under monitoring such as pressure, temperature, locations of moving objects, are much prone to uncertain data. For example any sensor network having limitations such as individual sensors battery power and network bandwidth limits the database system that stores the sensor data from containing exact values at all time. The measurement error incurred by the sensor further aggravates the problem and if these values are directly used for queries erroneous answers may result. The nature of uncertainty varies and depends on the application domain as in following examples [2].

Example1. In automated data cleaning applications there exists more than one alternative for the correct value. The selection of one among them may lead to incorrectness.

Example2. An example project at Purdue University that records movement of nurses as they carry RFID tags in the vicinity of the hospital. The readers installed in and outside the building track the presence of tags in their range. The variability of detection range and simultaneous detection of same tag by multiple readers incur difficulties in choosing a single location entity for a nurse at all times. To conclude, Data uncertainty arises naturally in many applications due to various reasons including measurement errors, data staleness, and repeated measurements.

To solve the uncertainty problem one of the most popular classification model is the decision tree model, as they are practical and easy to understand. Algorithms, ID3 and C4.5 devised for decision tree construction are widely adopted and used in a wide range of applications such as image recognition, medical diagnosis, credit rating of loan applicants, scientific tests, fraud detection, and target marketing. Due to uncertainty problem, The value of a feature/attribute is thus best captured not by a single point value, but by a range of values giving rise to a probability distribution. To handle uncertainty problem, consider the complete information approach carried by the probability distributions to build a decision tree, referred as Distribution-based approach. In section 2 we discussed the work done on similar issues. Section 3 provides the overview of the problem under consideration. Section 4 explains the proposed technique and generation of the decision tree. In section 5 we provide empirical values and observations of the experiments conducted on several real time data sets.

2. RELATED WORK

In recent years many new techniques for collecting data have resulted in an increase in the availability of uncertain data. While many applications lead to data which contains errors, we refer to *uncertain data sets* as those in which the level of uncertainty can be quantified in some way. Data uncertainty has been broadly classified as existential uncertainty and value uncertainty. Existential uncertainty appears when it is uncertain whether an object or a data tuple exists. For example, a data tuple in a relational database could be associated with a probability that represents the confidence of its presence.

Value uncertainty, on the other hand, appears when a tuple is known to exist, but its values are not known precisely. A data item with value uncertainty is usually represented by a

pdf over a finite and bounded region of possible values [3], [4]. The most general model for uncertain data is the *possible world's model* [1], which tries to capture all the possible states of a database which are consistent with a given schema. Sophisticated model for uncertain data is probabilistic graphical models which can be used in order to model complex dependencies. There has been a growing interest in uncertain data mining. In [4], the well-known k-means clustering algorithm is extended to the UK-means algorithm for clustering uncertain data. Data uncertainty is usually captured by pdf's, which are generally represented by sets of sample values. Mining uncertain data is therefore computationally costly due to information explosion (sets of samples vs. single values). To improve the performance of UK-means, pruning techniques have been proposed. Apart from studies in partition-based uncertain data clustering, other directions in uncertain data mining include density-based clustering. Density-based classification requires that the joint probability distribution of the data attributes be known. Decision tree classification on uncertain data has been addressed for decades in the form of missing values [5], [6]. Missing values appear when some attribute values are not available during data collection or due to data entry errors. Solutions include approximating missing values with the majority value or inferring the missing value (either by exact or probabilistic values) using a classifier on the attribute (e.g., ordered attribute tree and probabilistic attribute tree [7]).

In C4.5 [6] and probabilistic decision trees [8], missing values in training data are handled by using fractional tuples. During testing, each missing value is replaced by multiple values with probabilities based on the training tuples, thus allowing probabilistic classification results. a simple method of "filling in" the missing values could be adopted to handle the missing values, taking advantage of the capability of handling arbitrary pdf's in our approach. We present an approach to using graphical models for managing and querying large-scale uncertain databases. Another related topic is fuzzy decision tree. Fuzzy information models data uncertainty arising from human perception and understanding. The uncertainty reflects the vagueness and ambiguity of concepts, e.g., how hot is "hot". In fuzzy classification, both attributes and class labels can be fuzzy and are represented in fuzzy terms [11]. Given a fuzzy attribute of a data tuple, a degree (called membership) is assigned to each possible value, showing the extent to which the data tuple belongs to a particular value. Our work instead gives classification results as a distribution: for each test tuple, we give a distribution telling how likely it belongs to each class. There are many variations of fuzzy decision trees, e.g., fuzzy extension of ID3 [10] and Soft Decision Tree [9]. In these models, a node of the decision tree does not give a crisp test which decides deterministically which branch down the tree training or testing tuple is sent. Rather it gives a "soft test" or a fuzzy test on the point-valued tuple. Based on the fuzzy truth value of the test, the tuple is split into weighted tuples (akin to fractional tuples) and these are sent down the tree in parallel. As the volume of uncertain data increases, the cost of evaluating queries over this data will also increase some of the recent ideas for indexing uncertain data in support of range, nearest neighbor, and join queries[2]. These indexes build on standard well-known indexes such as R-trees and/or signature trees *(INDEXING UNCERTAIN DATA)*

3. PROBLEM STATEMENT

In this section, we mainly concentrate on handling of uncertain data for classification and how this uncertain data tuples are going to be handled.

3.1 Decision Tree

In traditional decision tree model which handles precise or single valued data; consists of d tuples like $\{t_1, t_2, t_3, \ldots, t_d\}$ and k feature attributes $\{A_1, A_2, \ldots, A_k\}$. The domain of each attribute A_j represented as $dom(A_j)$. Each tuple t_i is associated with a feature vector as $V_i = (V_{i,1}, V_{i.,2}, \ldots . V_{i,k})$ and class label as C_i. In this paper we consider binary decision trees with conditional tests on numerical valued attributes at each internal node n. An attribute Ajn that belongs to the training set of tuples and split point $z_n \in dom(A_{jn})$ are associated with each of these nodes and performing a binary test. . Each leaf node m in the decision tree is associated with a discrete probability distribution Pm over a set of class labels C. For each class label $c \in C$ a probability Pm(c) represents how likely a tuple would have a class label c associated with leaf node m. Determination of the class label of a given test tuple t0, the traversal starts from the root node until a leaf node is reached. At all the internal nodes visited a conditional test is executed and proceeds to the left or right child based on the test result. Eventually at a leaf node m we consider the class label $c \in C$ that maximises Pm(c).

3.2 Handler Uncertain Data

In uncertainty model an attribute value is represented by a probability density function(pdf) $f_{i,j}$ which is non zero only with in bounded interval $[a_{i,j};b_{i,j}]$. This closed form of pdf makes it feasible to be programmed analytically. It would be done numerically by having sample points $x \in [a_{i,j};b_{i,j}]$ with $f_{i,j}(x)$ as its associated value, approximating $f_{i,j}$ by a discrete distribution with s possible values effectively. This introduces large amount of processing cost as the availability of information increases by a factor of s. In our uncertainty model the decision tree classifies test tuples that contain uncertain attributes and its feature vector consists of pdf's $(f_{0,1},\ldots\ldots,f_{0,k})$. Our classification model maps such a feature vector to a probability distribution (P) over a set of class labels C. During this process each intermediate tuple t_x is associated with a weight $w_x \in [0; 1]$. We find conditional probability $\emptyset_n(c;t_x,w_x)$ recursively saying t_x has class label c by considering the sub tree rooted at n is used as an uncertain decision tree to classify intermediate tuple t_x with weight w_x. Including the root node we first check the attribute A_{jn} and split point z_n of node n to determine the $\emptyset_n(c;t_x,w_{x)}$.

4. PROPOSED METHOD

In this section, we discuss two approaches namely Averaging and Distribution-based techniques. We show that distribution-based technique improves the accuracy over averaging.

4.1 Averaging (AVG)

In this method, we transform uncertain dataset to a point-valued one by replacing each feature values with its mean value. More specifically, for each tuple t_i and attribute A_j, we take the mean value1

$$v_{i,j} = \int_{a_{i,j}}^{b_{i,j}} x f_{i,j}(x)\, dx$$

The feature vector of t_i is transformed to $(v_{i,1}, \ldots, v_{i,k})$. With this strategy, decision tree can be constructed using traditional decision tree algorithm.

4.2 Distribution-based

Using this approach, we can exploit full information of feature values which considers all the sample points that constitute each feature value. For decision tree construction, we can use same decision tree building framework as described above for handling point data. For node n, once we choose attribute A_{jn} and split node Zn, we have to split the set S into two subsets L and R. The major difference lies from point data case is the way set S is split

Recall that the pdf of a tuple $t_i \in S$ under attribute A_{jn} spans the interval $[a_{i,jn}, b_{i,jn}]$. If $b_{i,jn} \le z_n$, the feature values of t_i lies completely on the left of the split point and thus t_i is assigned to L and if $z_n \le a_{i,jn}$ then t_i is completely on the right side R. If the feature values are properly contains the split point, (i.e. $a_{i,jn} < z_n < b_{i,jn}$), we split t_i into two fractional tuples t_L and t_R in the same way as described below and add them to L and R, respectively as shown in fig 1. We call this algorithm UDT (for Uncertain Decision Tree).

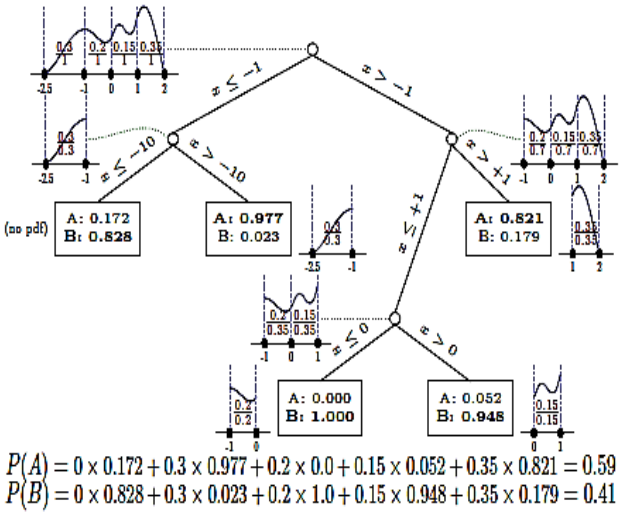
4.3 Construct Decision Tree

Decision tree for uncertain data reminds point-data model. With only difference is the way tree is employed to classify unseen test tuples. Here, test tuple t_0 contains uncertain attributes, that is, it's feature vector is thus a vector of feature values $(f_{0,1}, \ldots, f_{0,k})$. Thus a classification model is a function M that maps such a feature vector to probability distribution *P* over *C*.

The probabilities for P are calculated as follows:

- During construction, we associate each intermediate tuple t_x with a weight $w_x \in [0, 1]$. Further, we recursively define the quantity $\emptyset_n(c;t_x,w_x)$, which is interpreted as the conditional probability that t_x has class label c, with sub tree rooted at n is used as an uncertain decision tree to classify tuple t_x with weight w_x.
- Including the root, for each internal node n to determine $\emptyset_n(c;t_x,w_x)$, first we check the attribute A_{jn} and split point z_n of node n.
- Left probability calculated as $P_L = \int_{ax,jn}^{Zn} f_{x,jn}(t)\,dt$ (or P_L=0 incase $Z_n < a_{x,j_n}$) and right probability is $P_R = 1 - P_L$. Then, we split t_x into 2 fractional tuples t_L and t_R. Each tuple inherits the class label of t_x as well as the feature value of t_x for all attributes except A_{jn}.
- Weight for tuple t_L becomes $w_L = w_x \cdot p_L$ and its feature value for A_{jn} is given by
  $f_{L,jn}(x) = f_{x,jn}(x)/w_L$ if $x \in [a_{x,jn}, z_n]$;
  0 otherwise
- Weight and feature value for tuple T_R assigned analogously to the left tuple.

Finally, quantity for node becomes $\emptyset_n(c;t_x,w_x) = P_L.\emptyset_{nL}(c; t_L,w_L) + P_R.\emptyset_{nR}(c; t_R,w_R)$ where n_L and n_R are the left child and the right child of node n, respectively.



$$P(A) = 0 \times 0.172 + 0.3 \times 0.977 + 0.2 \times 0.0 + 0.15 \times 0.052 + 0.35 \times 0.821 = 0.59$$
$$P(B) = 0 \times 0.828 + 0.3 \times 0.023 + 0.2 \times 1.0 + 0.15 \times 0.948 + 0.35 \times 0.179 = 0.41$$

Fig 1: UDT classifying test tuples.

5. EMPIRICAL RESULTS

To explore the potential of achieving higher classification accuracy for uncertain data, we have implemented AVG and UDT and applied them to different real data sets. Taken from the UCI Machine Learning Repository [12].

The algorithms described above have been implemented in Java using JDK 1.6 and a series of experiments were performed on a PC with an Intel i5 2.66GHz CPU and 8GB of main memory, running Windows 7.

Below Table1 shows the performance difference in terms of accuracy between AVG and UDT where UDT resulted in more accuracy. As we have mentioned before the computational cost of UDT is more comparatively. In Fig.2 We plotted the accuracy measures of both UDT and AVG while Fig.3 depicts the variation in the computational costs.

TABLE1: ALGORITHM RESULTS

			Accuracy %		Time(sec's)	
Dataset	Records	Dimension	AVG	UDT	AVG	UDT
Mushroom	8124	90	96.12	98.6	0.3	0.4
Nursery	12960	32	95.32	96.84	0.5	0.5
PageBlocks	5473	46	93.56	92.91	0.2	0.2
PimaIndians	768	46	86.34	92.71	0.2	0.2
Soybean	683	118	91.98	93.81	0.2	0.3
TicTacToe	958	29	95.77	98.96	0.2	0.3
Waveform	5000	101	94.22	96.44	0.7	0.8
WeatherData	14	12	97.28	99.91	0.1	0.1
Wine	178	68	98.21	99.52	0.2	0.2
Zoo	101	42	98	99.99	0.1	0.1

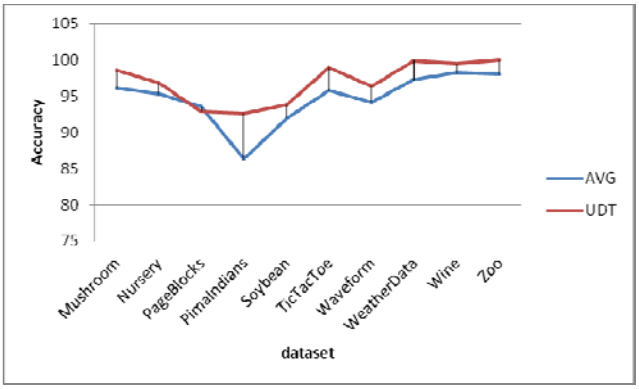


Fig 2: Accuracy graph

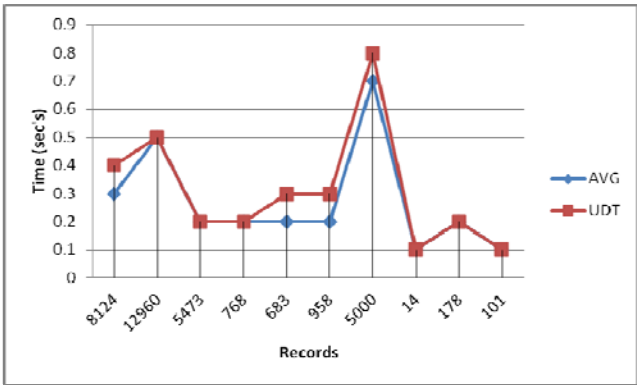


Fig 3: Time complexity graph

6. CONCLUSION

We have proposed the new technique by enhancing the traditional decision tree algorithm in order to classify the range bound values of features. The empirical results are provided in support of our proposal. With this we conclude that UDT classified the tuples with range bound values more accurately than existing averaging technique by sacrificing computational cost.

REFERENCES

[1] MANAGING AND MINING UNCERTAIN DATA Edited by CHARU C. AGGARWAL IBM T. J. Watson Research Center, Hawthorne, NY 10532. http://charuaggarwal.net/utoc.pdf

[2] Sarvjeet Singh, Chris Mayfield, Sunil Prabhakar, Rahul Shah and Susanne Hambrusch. Indexing Uncertain Categorical Data. In IEEE 23rd International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey, April 2007. [Paper]

[3] J.Chen and R. Cheng, "Efficient evaluation of imprecise location dependent queries," in ICDE. Istanbul, Turkey: IEEE, 15-20 Apr. 2007, pp. 586–595.

[4] M. Chau, R. Cheng, B. Kao, and J. Ng, "Uncertain data mining: An example in clustering location data," in PAKDD, ser. Lecture Notes in Computer Science, vol. 3918. Singapore: Springer, 9–12 Apr. 2006, pp. 199–204.

[5] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.

[6] C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993, ISBN 1- 55860-238-0.

[7] L. Hawarah, A. Simonet, and M. Simonet, "A probabilistic approach to classify incomplete objects using decision trees," in DEXA, ser. Lecture Notes in Computer Science, vol. 3180. Zaragoza, Spain: Springer, 30 Aug.- 3Sep. 2004, pp. 549–558.

[8] J. R. Quinlan, "Learning logical definitions from relations," Machine Learning, vol. 5, pp. 239–266, 1990.

[9] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," Fuzzy Sets and Systems, vol. 138, no. 2, pp. 221–254, 2003.

[10] M. Umanol, H. Okamoto, I. Hatono, H. Tamura, F. Kawachi, S. Umedzu, and J. Kinoshita, "Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems," in Fuzzy Systems, 1994, vol. 3. IEEE World Congress on Computational Intelligence, 26-29 Jun. 1994, pp. 2113–2118.

[11] Y. Yuan and M. J. Shaw, "Induction of fuzzy decision trees," Fuzzy Sets Syst., vol. 69, no. 2, pp. 125–139, 1995.

[12] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online].
Available: http://www.ics.uci. edu/_mlearn/MLRepository.html