

Classification of Categorical Uncertain Data Using Decision Tree

Shweta S. Thakur
Computer department
PG Student MET BKC IOC
Nashik, India
E-mail: thakurshweta2228@gmail.com

Abstract: Certain data is a data whose values are known precisely whereas uncertain data means whose value are not known precisely. But data is always uncertain in real life applications. In data uncertainty attribute value is represented by a set of values. There are two types of attributes in data sets namely, numerical and categorical attributes. Data uncertainty can arise in both numerical and categorical attributes. Traditional decision tree algorithms work with certain data only. The classification performance of decision tree can be improved if complete information of data is considered. Probability Density Function (PDF) is used to improve the accuracy of decision tree classifier. Existing system to handle uncertain data works on only numerical attributes means only range of values. They cannot work on uncertain categorical attributes. This paper proposes a method for handling data uncertainty on categorical attributes. The decision tree algorithm is extended to handle uncertain data. The experiments show that the classification performance of this decision tree can be enhanced.

Keywords: Data Mining, Decision Tree, Uncertain Data, Probability density Function, Classification.

I. INTRODUCTION

Classification of large data set is an important problem in machine learning and data mining. Data mining technique refers to extracting or mining knowledge from large data. Database contains number of training tuples and a set of classes such that each tuple belongs to one of the given class. The problem of classification is that to decide the class to which a given record belongs. The classification problem is also concerned with generating a description or a model for each class from the given dataset. Classification is used to classify unseen test tuple with high accuracy. For real world application construction of classifiers should be fast.

There are different kinds of classification models such as decision tree, Bayesian classifier, rule-based classifier, support vector machines (SVM), artificial neural network and lazy learners. Decision tree is one of the most popular classification models. Decision tree [1] are popular because it is practical and easy to understand. Decision tree structure contains an internal nodes, leaf nodes and branches. The top node in tree is defined a root node. For decision tree construction many algorithms like SLIQ, ID3, C4.5 [2] and SPRINT, CART have been devised. All these algorithms are used in a wide range of applications such as image recognition, medical diagnosis, credit rating of loan application, scientific taste, fraud detection and target marketing. The construction of Decision trees [3] does not require any domain knowledge

In traditional decision tree classification, an attribute of a tuple is either numerical or categorical. The feature which works on numerical data known as numerical

feature and the feature whose domain is not numeric are called the categorical feature. The aim of classification is to predict the class of a tuples whose class label is unknown. Data uncertainty is common in many applications. The classification performance of decision tree can be improved if complete information of data is considered. To abstract probability distribution by summary statistics such as mean and variance is a simple way to

handle data uncertainty this approach is known as averaging. Another method is to work on complete information carried by the probability distribution to build a decision tree this method is known as distribution-based method.

Data uncertainty comes in many applications due to various reasons. The different kinds of uncertainty [5] are shortly discussed here:-

- 1) Parameter uncertainty arises from the model parameter which is input to the computer but whose exact value are unknown to the experimentalist and cannot be controlled in physical experiment.
- 2) Structural uncertainty comes from lack of knowledge of underlying true physics and it depends on how correctly a mathematical model describes a true system for a real situation, in consideration of the fact that models almost give the approximation of the real situation.
- 3) In case of variability of experimental measurement experimental uncertainty arises. The experimental uncertainty is inevitable and may be detected by repeating a measurement for several times using exactly same setting for all inputs.

From the above examples, it has been noted that in several applications information cannot be ideally represented by point value. It is best captured by a range value with a PDF.

II. LITERATURE SURVEY

There has been a growing interest in uncertain data management in recent years. Data uncertainty has been classified into three types [1] –

- 1) Existential or tuple uncertainty
- 2) Value uncertainty or attribute uncertainty
- 3) Co-related uncertainty.

When uncertain object or the uncertain data tuples exists then existential uncertainty appears. It is also arises due to uncertain a feature of a data objects. Data uncertainty is a “probabilistic database”, in that each data tuples is

situated with a probability value which contains the confidence of its presence [6]. Value uncertainty appears when a tuple is known to exist, but the tuple value is not known precisely. In data item value uncertainty is usually represented by a PDF over a finite and the bounded region of possible values [12]. "Imprecise queries processing" is one well known topic on the value uncertainty. Such a query is associated with a probability that represent the guarantee on its correctness. In co-related uncertainty value of multiple attributes describe by a joint- probability- distribution

Significant research has been interested in uncertain data mining [8] in recent years. Goal of the data mining process is to extract information from data set and transform this data into an understandable format for further use. For the uncertain data mining, extend traditional data mining algorithm in such a way that the extended data mining algorithm are applicable for uncertain data. The extension process contains two main steps to allow traditional data mining algorithm computationally feasible for uncertain data [9].

To convert the traditional data mining algorithm to make it theoretically workable for uncertain data is the first step. For clustering uncertain data well-known k-means clustering algorithm is extended to the uk-means algorithm [10]. In traditional clustering algorithm, data is generally represented by points in the space. To improve the I/O time and computational efficiency of the modified algorithm is the second step. In clustering, data uncertainty is generally captured by PDF and usually represented by sets of sample values. The uncertain data mining is therefore computationally costly due to the information explosion. To improve the efficiency of K-means and UK-means series of pruning technique have been proposed. Examples include ck-means [11] and min-max-dist. pruning.

The decision tree classification has been addressed for several decades. Decision tree is one of the most important algorithms for "Decision- making". Decision tree are valuable tools when it comes to the description, classification and generalization files. Decision tree are mainly used for classification. Decision tree are popular because of its interpretable- decision tree can easily converted into a set of if- then rules that are easy to understand. In real life most of the data mining problems found in the classification. Decision tree classification is one of the best solution approaches. Quinlan proposed ID3 and C4.5 algorithms, which is elegant and instinctive solution. Different dispersion measures technique has been introduced to measure the goodness of split for decision tree. Popular dispersion measure includes Gini index, information gain and gain ratio. Gini index is used in a CART [7] and information gain is used in ID3 as dispersion measure. The gain ratio is the extension of information gain and used in C4.5 [2]. A series of pruning technique are introduced to deal with the problem of over fitting the data. Pruned decision tree are faster in classifying unseen test tuples because pruned decision tree are smaller than unpruned decision tree. There are two techniques of pruning a decision tree namely – pre-pruning and post-pruning. Pre-pruning technique stops the construction of decision tree earlier. On other hand post-pruning technique removes branches from fully constructed decision tree.

Classification of uncertain data has been studied for decades in the form of missing values. Missing value appears when some attribute value is not available during data collection process or due to data entry error. For handling missing data methods are divided into three category containing, ignoring or discarding data, parameter estimation and imputation. Missing value affect the accuracy of the classifier, so that proper handling of it is important. One of the ways to handle the missing value is to ignore the tuple. Another way is to use the pattern of other data tuples to approximate missing values. Simplest way is to use the majority values or most commonly known values to approximate missing values [4].

Fuzzy decision tree is another related topic in decision tree classification. Fuzzy data models data uncertainty arising from human perception and understanding. In real world Fuzzy data is reflecting more accurately. Data which used in many real world application are inherently impress and subjective in nature. When expert tries to analyze certain event or fact, him or her express the fact with some degree of surety. These facts are represented in terms of logical and numerical values. Use of multi-valued logic technique does not always give necessary reliance. Numerical values search common characteristics from input attribute but presence of variety of real values. This technique is more difficult. Rather use any frequency criteria to distinguish different data, people use their subjective feelings, short-time memory and background knowledge. Fuzzy decision tree is a best approach to capture this vagueness of information.

Example: How cool is, "cool ". In fuzzy decision tree, attribute and class label can be fuzzy and they are represented in fuzzy terms. Fuzzy attribute of a data tuple are given, a degree is assigned to each possible value. They show the extension to which the data tuple belongs to particular value. There are different types of fuzzy decision tree containing, soft decision tree and fuzzy extension of ID3.

III. METHODOLOGY

Uncertain data can be handled by using following two approaches. The first approach is called as "averaging" which transforms an uncertain data set into a point value by replacing each PDF with its approximate mean value. Mean value is represented by,

$$v_{i,j} = \int_{a_{i,j}}^{b_{i,j}} x f_{i,j}(x) dx \quad \dots (1)$$

By applying a traditional tree construction algorithm decision tree can be constructed. The second approach called "distribution-based" approach can exploit the full information carried by PDF by considering all sample points that constitute each PDF [1].

In the following section details of the tree construction algorithm under “Averaging” and “distribution-based” approaches are presented.

A. Averaging

In existing decision tree classifier a dataset consist of i^{th} training tuple is associated with a set of values of numerical attributes of training data sets and i^{th} training tuple is represented as, $t_0 = (t_{i,1}, t_{i,2}, t_{i,3}, \dots, t_{i,k}, Classlabel)$ Where,

- i = Tuple number and k is the number of numerical attributes in that training data set.

- $t_{i,1}$ = Value of first attribute of i^{th} tuple.

- $t_{i,2}$ = Value of second attribute of i^{th} tuple.

This approach is called as AVG (for Averaging). It is required to determine the class label of a given test tuple t_0 , we traverse the decision tree from root node to a specific leaf node. Test tuple t_0 is represented as,
 $t_0 = (a_1, a_2, a_3 \dots a_k, ?)$

Decision tree classifier is constructed by splitting each node into left and right nodes. Initially, the root node contains all training tuples. The training data tuples in node is partitioned in to two nodes based on the best split point value z_n of the best split attribute A_j and result is stored in its left and right nodes, it is referred to as splitting. Whenever further split of node is not required then it is called as leaf node.

Entropy is a function or metric which is used to find the degree of dispersion of training data tuples in a node. To build a good decision tree is depends on the choice of A_j and z_n . At this point we assume this selection is performed by Best Split black box algorithm. Best Split algorithm takes a set of tuples as parameter, and returns the best choice of split point and attribute for those tuples. In decision tree construction the goodness of a split is measured by an impurity measure [2]. The degree of dispersion can be measured in many ways, namely entropy and Gini index. In this paper we consider entropy is used as the dispersion measure. Entropy is based on the proportions of tuples of each class in the training data set and it is an information based measure. Entropy is predominantly used for constructing decision trees so that it is taken as dispersion measure.

Entropy is calculated using following formula,

$$Entropy(s) = \sum_{i=1}^m P_i \cdot \log_2(P_i) \dots (2)$$

Where,

P_i = Probability of number of tuples belonging to the i^{th} class.

$$H(Z, A_j) = \sum_{X=L,R} \frac{|X|}{|S|} \left(\sum_{c \in C} \frac{P_c}{X} \log_2 \left(\frac{P_c}{X} \right) \right) \dots (3)$$

$$H(Z, A_j) = \frac{|L|}{|S|} (Entropy(L)) + \frac{|R|}{|S|} (Entropy(R)) \dots (4)$$

Where,

- A_j = Splitting attribute.

- L = Total number of tuples to the left side of the split point z .

- R = Total number of tuples to the right side of the split point z .

- S = Total number of tuples in the node.

B. Distribution-based on Numerical Data:

The procedure for creating Decision Tree classifier on Uncertain Data is same as that of Decision Tree classifier on Certain Data except that Decision Tree on Uncertain Data use probability density function with equal probabilities to calculates entropies for all modeled data values of the numerical attributes of the training data sets. This approach is called as UDT (for Numerical Uncertain Decision Tree) [1].

For each value of the numerical attribute, an interval is created and a set of ‘n’ sample values are generated within the interval. This is obtained using probability density function and Gaussian distribution with attribute value as the mean and standard deviation. Entropies are calculated for all ‘n’ sample points within that interval and finally point with minimum entropy is selected.

The UDT algorithm constructs a decision tree by splitting each node into left and right nodes. Initially, the root node contains all the training data tuples. Using probability density function model for each value of numerical attribute in the training dataset a set of ‘n’ sample values are generated and then stored in the root node. Entropy values are calculated for split point’s k ($mn - 1$). Where,

k = Number of numerical attributes in training data set.

m = Number of training data tuples at current node T

n = Number of probability density function values for each numerical attribute in the training data set.

The process of partitioning the training data tuples in a node into two subsets is same as that of Averaging algorithm.

IV. PROPOSED METHODOLOGY

In existing methodology we have been focusing on processing uncertain numerical attribute but uncertainty can also arises in categorical attributes. The sources of uncertainty in categorical attributes include ambiguities, data staleness and repeated measurement. The different kinds of uncertainty are shortly discussed here:-

- 1) Users are cluster on access logs of HTTP proxy server by using top level domain name, “.org”, “.edu”, “.com”, “.in”, “.co”, “.jp”, “.de”. Consider an attribute from multiple log entries generated by each user we obtain repeated measurement of this attribute. The multiple value of this attribute collected from these entries, which naturally describes the uncertainty arises in categorical attribute.
- 2) At the time of recording the color of a traffic light signal is red, could have changed to green or yellow in 10 seconds. This is an example of uncertainty arises in data staleness.
- 3) In a survey color of flowers may divide into a number of categories, which may overlap with one another this is called as an ambiguities could be recorded as a distribution.

A. Distribution-based on Categorical data:(C_UDT)

Build a decision tree on uncertain categorical attribute in top-down manner, starting from root node. For a tuple t_i with uncertain attribute A_j , the value uncertainty can be modeled by a discrete probability distribution function $f_{i,j}$: $\text{Dom } A_j \rightarrow [0, 1]$ satisfying $\sum_{x \in \text{dom}(A_j)} f_{i,j}(x) = 1$. In decision tree an internal node n corresponding to a categorical attribute A_j is not associated with a split point. Internal node n has many child nodes, each corresponding to a distinct value in $\text{Dom } A_j$. At each node, all possible categorical attribute are considered. Entropy is calculated for attribute and attributes which give high information gain is selected. For building a tree, node is assigned that attribute and the tuples are propagated to the child nodes.

Calculate the entropy of categorical attribute A_j , split the tuples into a set of buckets $\{B_v | v \in \text{dom}(A_j)\}$. Tuple t_x is copied into bucket B_v as new tuple t_y where weight is $w_y = f_{x,j}(v)$ and $w_y > 0$. The probability density function of t_y are inherited from t_x , except for attribute A_j . The value of attribute is set to $f_{y,j}(w) = 0$ and $f_{y,j}(v) = 1$ for all $w \neq v$. Using all the buckets entropy for the split on attribute A_j is calculated. As a heuristic, if categorical attribute is already chose for splitting in an ancestor node of the tree then there is no need to reconsider it, because it will not give any information gain if the tuples in question are split on that categorical attribute again. To check the value of attribute A_j in test tuple the test is perform at node n and

the action taken is to follow the branch to the child node corresponding to that attribute value.

C_UDT Algorithm:

Uncertain_Decision_Tree on Categorical Attributes (T):

Input: The training dataset D (uncertain categorical dataset); the set of candidate attributes att-list.

Output: An uncertain decision tree.

Begin

1. create a node N;
2. if (dataset D have Tuples T and all of the same class, C) then
3. return Node(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 dataset D
6. end if;
7. If tuples in the node T have more than one class then
8. Find_Categorical_Best_Split(T)
9. Node n assigned to attribute
10. for $i \leftarrow 1$ to data size [T] do
11. if split_attribute_value [] \leq split Criteria[T] then
12. Put tuple in to left [T],
13. else
14. Put tuple in to right [T]
15. end if ;
16. If left [T] = NULL or right[T] = NULL then
17. Create empirical probability distribution of the node
18. Return (T)
19. If left[T] != NULL and right[T] != NULL then
20. DECISION_TREE (left [T])
21. else
22. DECISION_TREE (right [T])
23. Return (T)
24. end if;
25. end for ;

V. RESULTS AND DISCUSSION

Data sets are taken from University of California Irvine repository. Experiments were performed on these data sets to obtain performance of the proposed distribution based on categorical approach. The algorithms describe above are implements in java using JDK 1.6 and Net Beans IDE. The environment use for experiments includes a PC with an Intel Core 2 Duo 2.66GHz CPU and 2GB main memory. The results of these experiments are compared with certain decision tree.

Here we can handles categorical uncertain data which is never exists before. Here modification is done in existing algorithm UDT. It shows that, C_UDT approach handle data uncertainty on categorical attribute and it compare with certain decision tree (J48).

Table 1: Data sets from the UCI Machine Learning Repository

No	Dataset name	Total tuple	No. of Attribute	No. of Classes
1	Car	1728	6	4
2	Nurse	12960	8	5
3	Chess	3196	36	2
4	Balloon	20	4	4
5	Soybean	307	35	19

The classifier accuracy is determined by the percentage of the test data set that is correctly classified. Then compare the accuracy of certain decision tree (J48) and C-UDT, in result we get accuracy of C-UDT is improve.

Table 2: Accuracy comparison of certain decision tree(CDT) and uncertain decision tree (C-UDT)

No	Dataset name	Total tuple	Accuracy of certain decision tree (J48)	Accuracy C-UDT
1	Car	1728	92.36	94.45
2	Nurse	12000	98.32	98.45
3	Chess	3196	50.58	97.05
4	Balloon	1562	100	89.56
5	Soybean	307	91.5	96.05

Fig.1. Shows the accuracy comparison graph of J48 and C-UDT where x-axis defines accuracy in percentage and y-axis defines different types of data sets.

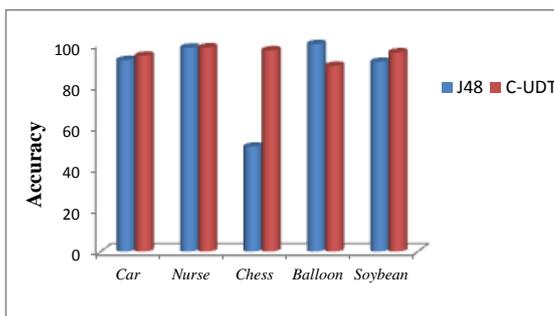


Figure1. Accuracy comparison of CDT and C-UDT

VI. CONCLUSION

This paper extended the model of traditional decision tree classification to accommodate data tuples having categorical attributes with uncertainty described by arbitrary probability density function (PDF). The decision tree algorithm is

extended to handle uncertain data. The Distribution-Based on categorical approach is compared with existing approach. It is observed that decision tree build from C_UDT is more accurate. However, the computation of probability density function makes the classifier more accurate. The experiments show that the classification performance of this decision tree can be enhanced.

REFERENCES

- [1] Smith Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho, and Sau Dan Lee, "Decision Trees for Uncertain Data", IEEE Transaction on Knowledge and Data Engineering, VOL. 23, NO. 1, January 2011.
- [2] J. R. Quinlan, "C4.5: Program for machine learning" Morgan Kaufman, San Mateo,1993
- [3] D. V. Patil & R. S. Bichkar "Issues in Optimization of Decision Tree Learning: A Survey" "International Journal of Applied Information Systems – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 3 – No.5, July 2012.
- [4] Miss Pragati Pandey& Mrs. Minu Choudhary "Uncertain Data Management and Mining" International Journal of Computer Science and Information Technology & Security, ISSN: 2249-9555Vol. 2, No.6, December 2012.
- [5] VarshaChoudhary & Pranita Jain "Classification: A Decision Tree For Uncertain Data Using CDF" International Journal of Engineering Research and Applications ISSN: 2248-9622 Vol. 3, Issue 1, January -February 2013.
- [6] Nilesh N. Dalvi and Dan Suciu. "Efficient query evaluation on probabilistic databases" The VLDB Journal, 16(4):523–544, 2007.
- [7] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. "Classification and Regression Trees" Wadsworth, 1984.
- [8] Michael Chau, Reynolds Cheng, Ben Kao, and Jacky Ng. Uncertain data mining: An example in clustering location data. In PAKDD, volume 3918 of Lecture Notes in Computer Science, pages 199–204, Singapore, 9–12 April 2006. Springer.
- [9] P. SatyaPrakash, P. Jhansi Lakshmi and B.L Krishna "Developing Decision Trees for Handling Uncertain Data" International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012,4480 – 4485.
- [10] M. Chau, R. Cheng, B. Kao, and J. Ng, "Uncertain Data Mining: An Example in Clustering Location Data," Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, pp. 199-204, Apr.2006.
- [11] Sau Dan Lee, Ben Kao, and Reynolds Cheng. Reducing UK-means to K-means. In The 1st Workshop on Data Mining of Uncertain Data (DUNE), in conjunction with the 7th IEEE International Conference on Data Mining (ICDM), Omaha, NE, USA, 28 October 2007.
- [12] Jinchuan Chen and Reynolds Cheng. Efficient evaluation of imprecise location dependent queries. In Proceedings of the 23rd International Conference on Data Engineering, pages 586–595, Istanbul, Turkey, 15-20 April 2007. IEEE.