ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

DECISION TREES FOR UNCERTAIN DATA BASED ON STATISTICAL UNIFORM DISTRIBUTION

C. SUDARSANA REDDY[1], Dr. V. VASU[2], S. AQUTER BABU[3]

Department of Computer Science and Engineering, S.V. University College of Engineering, S.V. University, Tirupati, Andhra Pradesh, India[1]

Department of Mathematics, S.V. University, Tirupati, (A.P), India[2]

Assistant Professor of Computer Science, Department of Computer Science, Dravidian University, Kuppam -517425, Chittoor District, Andhra Pradesh, India[3]

Abstract: Certain or classical decision trees are constructed for training data sets containing certain data. But in real life, in many cases, data is always uncertain. Hence many previous data mining techniques such as classification, clustering, regression and association rule mining etc. are inefficient or inaccurate or they must be reconsidered in managing uncertain data. Present study proposes an efficient and more accurate uncertain data management technique in data classification using decision trees. This new technique is modelled using uniform distribution and it is called Uniform Decision Trees for Uncertain Data (UDTUD). Uniform decision tree classifiers constructed for uncertain data are more accurate than Certain Decision Tree (CDT) classifiers constructed using certain data. There exists many models for uncertain data management but we propose Uniform distribution model for uncertain data management because it gives more accurate results for some training data sets. Applying data mining techniques to uncertain data is computationally costly. Extensive experiments have been conducted which show that classification accuracies obtained by UDTUD are more accurate than classification accuracies obtained by Certain Decision Trees (CDTs).

Keywords: Uniform distribution, uncertain data, certain data, decision tree, classification, data mining, machine learning

1. INTRODUCTION

Decision tree induction is the learning of decision trees from class-labeled training tuples [1]. The task of constructing a decision tree from the training data set is called decision tree induction [2]. Data uncertainty arises in many applications because of measurement errors, data staleness and repeated measurements [3]. Decision trees have been well recognized as very powerful and attractive classification tools [4].

Data mining applications for uncertain data are – classification, clustering, frequent pattern mining, and outlier detection etc. Attributes in the training data sets are of two types: categorical and numerical. Data uncertainty exists in both categorical and numerical attribute values. Data uncertainty is divided into two types: existential uncertainty and value uncertainty. In existential uncertainty a tuple may or may not exist in the relation. In the case of value uncertainty there exist a tuple in the relation but values of attributes in the tuple may or may not exist [3].

In many cases, values of attributes are inaccurate or approximate. Sometimes values of an attribute are specified as a range. Attribute value may take any value in the range. There may be many possible values of an attribute within the range, but which single value is the correct value? Given representative value of an attribute it is modified or accurately estimated by a Uniform distribution. Experimental results show that decision trees constructed for Uniform estimated values are more accurate than certain decision trees constructed using given representative values.

Data uncertainty means range of values for numerical attributes and set of values for categorical attributes [3]. Various sources of data uncertainty in the values of attributes in the training data sets are – repeated measurements, measurement problems, quantization, continuously data changing, sometimes data itself contains fuzzy features, and privacy preserving of data. We propose a new method of handling data uncertainty using Uniform distribution sampling technique. Frequently data uncertainty is modeled by Uniform distribution for digitization errors.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

1.1 PRUNING TECHNIQUE

For each given representative value of each attribute, Uniform distribution is applied and a set of best approximated new values are generated and entropy is calculated for each new value and then one best new point called optimal split point is selected corresponding to minimum entropy value.
To reduce the computational complexity of entropy it is enough to compute entropy only at one newly generated value through Uniform distribution but with a very small reduction in classification accuracy than the classification accuracy obtained when entropy is calculated for all the newly generated values of attributes using Uniform distribution.

2. INTRODUCTION TO UNCERTAIN DATA

With data uncertainty data values are no longer atomic or certain. Data is often associated with uncertainty because of measurement errors, sampling errors, repeated measurements, and outdated data sources. When data mining techniques are applied on uncertain data it is called Uncertain Data Mining (UDM). For preserving privacy sometimes certain data values

are explicitly transformed to range of values. For example, for preserving privacy the certain value of the true age of a person is represented a range of [26, 32] or 26 – 32.

Tuple No	Marks (Numerical)	Result (Categorical)	Class Label (Categorical)
1	550 – 600	(0.8,0.1,0.1,0.0)	(0.8,0.2)
2	222 – 444	(0.6,0.2,0.1,0.1)	(0.5,0.5)
3	470 – 580	(0.7,0.2,0.1,0.0)	(0.9,0.1)
4	123- 290	(0.4,0.2,0.3,0.1)	(0.7,0.3)
5	345 – 456	(0.6,0.2,0.1,0.1)	(0.8,0.2)
6	111 – 333	(0.3,0.3,0.2,0.2)	(0.9,0.1)
7	200 – 280	(0.3,0.3,0.2,0.2)	(0.7,0.3)
8	500 – 580	(0.7,0.2,0.1,0.0)	(0.5,0.5)
9	530 – 590	(0.7,0.3,0.0,0.0)	(0.6,0.4)
10	450 – 550	(0.7,0.2,0.1,0.0)	(0.4,0.6)

Table 1.1 Example of Numerical Uncertain and Categorical Uncertain attributes.

Marks are a numerical uncertain attribute (NUA) and Result and class label are categorical uncertain attributes (CUAs).

3. PROBLEM DEFINITION

In many real life applications information cannot be ideally by represented by point or certain data only. Data uncertainty was not considered during the development of many data mining algorithms including decision tree classification technique. All decision tree algorithms so far developed were based on certain data only. Data uncertainty was not considered during decision tree building step. Thus, there is no classification technique which handles the uncertain data. This is the problem with the existing certain (traditional or classical) decision tree classifiers.
Currently existing decision tree classifiers consider values of attributes in the tuples with known and precise point data values only. In real life the data values inherently suffer from value uncertainty (attribute uncertainty). Hence, certain (traditional or classical) decision tree classifiers produce incorrect or less accurate data mining results. As data uncertainty widely exists in real life, it is important to develop accurate and more efficient data mining techniques for uncertain data management. A training data set can have both uncertain numerical attributes (UNAs) and uncertain categorical attributes (UCAs) both training tuples as well as test tuples contain uncertain data.
The present study proposes an algorithm called Uniform Decision Tree for Uncertain Data (UDTUD) to improve performance of Certain Decision Tree (CDT). UDTUD uses Uniform distribution or Uniform error modeling technique to correct data uncertainty in values of numerical attributes of training data sets. The performance of these two algorithms is compared experimentally through simulation. The performance of UDTUD is proves to be better.

4. EXISTING ALGORITHM

4.1 Certain Decision Tree (CDT) Algorithm Description

The certain decision tree (CDT) algorithm constructs a decision tree classifier by splitting each node into left and right nodes. Initially, the root node contains all the training tuples. The process of partitioning the training data tuples in a node into two subsets based on the best split point value z_T of best split attribute A_{j_T} and storing the resulting tuples in its left and right nodes is referred to as splitting. Whenever further split of a node is not required then it becomes a leaf node referred to as an external node. All other nodes except root node are referred as internal nodes. The splitting process at each internal node is carried out recursively until no further split is required. Continuous valued attributes must be discretized prior to attribute selection [7]. Further splitting of an internal node is stopped if one of the stopping criteria given hereunder is met.

1.All the tuples in an internal node have the same class label
2.Splitting does not result nonempty left and right nodes.

In the first case, the probability for that class label is set to 1 whereas in the second case, the internal node becomes external node. The empirical probabilities are calculated for all the class labels of that node. The best split pair comprising an attribute and its value is that associated with minimum entropy. During decision tree construction within each internal node only crisp and deterministic tests are applied. One possible function to measure impurity is entropy [2]. Entropy is an information based measure and it

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

is based only on the proportions of tuples of each class in the training data set.

Accuracy and execution time of CDT algorithm for 9 data sets are shown in Table 6.2.

Entropy is calculated using the formula

$$entropy(S) = \sum_{i=1}^{m} -p_i . log_2(p_i)$$

Where p_i = number of tuples belongs to the i^{th} class

$$H(z, A_j) = \sum_{X=L,R} \frac{|X|}{|S|} \left(\sum_{c \in C} -\frac{p_c}{X} log_2 \left(\frac{p_c}{X} \right) \right)$$

$$H(z, A_j)$$

$$= \frac{|L|}{|S|} \left(\sum_{c \in C} -\frac{p_c}{L} log_2 \left(\frac{p_c}{L} \right) \right)$$

$$+ \frac{|R|}{|S|} \left(\sum_{c \in C} -\frac{p_c}{R} log_2 \left(\frac{p_c}{R} \right) \right) \qquad (4.1)$$

$$H(z, A_j) = \frac{|L|}{|S|} \big(Entropy(L) \big) + \frac{|R|}{|S|} \big(Entropy(R) \big)$$

Where

- A_j is the splitting attribute.
- L is the total number of tuples to the left side of the split point z.
- R is the total number of tuples to the right side of the split point z.
- $\frac{p_c}{L}$ is the number of tuples belongs to the class label c to the left side of the split point z.
- $\frac{p_c}{R}$ is the number of tuples belongs to the class label c to the right side of the split point z.
- S is the total number of tuples in the node.

4.2 Pseudo code for Certain Decision Tree (CDT) Algorithm

```
CERTAIN_DECISION_TREE (T)
1.    If all the training tuples in the node T have the
2.      same class label then
3.        set p_T(c) = 1.0
4.    return(T)
5.    If tuples in the node T have more than one class
then
6.    Find_Best_Split(T)
7.    For i ← 1 to datasize[T] do
8.    If split_atribute_value[t_i] <= split_point[T] then
9.       Add tuple t_i to left[T]
10.   Else
11.      Add tuple t_i to right[T]
12.   If left[T] = NIL or right[T] = NIL then
13.   Create empirical probability distribution of the
node T
14.      return(T)
15.   If left[T] != NIL and right[T] != NIL then
16.      CERTAIN_DECISION_TREE(left[T])
17.      CERTAIN_DECISION_TREE(right[T])
18.   return(T)
```

5. PROPOSED ALGORITHM

5.1 Proposed Gaussian Decision Tree (GDT) for Uncertain Data (GDTUD) Algorithm Description

The procedure for creating Uniform Decision Tree (UDT) is same as that of Certain Decision Tree (CDT) except that UDT calculates entropy values for uncertain data values in the numerical attributes of the training data sets by constructing intervals. Errors in the values of numerical attributes in the training datasets are calculated based on the assumption that data sets contain measurement errors particularly when the data sets contain numerical attributes.

For each value of each numerical attribute an interval is constructed and within the interval a set of 'n' sample values are generated using Uniform distribution with the attribute value as the mean and standard deviation as the length of the interval divided by 6 and then entropies are computed for all uncertain data values of 'n' sample points within that interval and the point with minimum entropy is selected. Based on the assumption that measurement errors are inevitable in the values of numerical attributes in the training dada sets, errors are corrected by using Uniform distribution in the values of numerical attributes.

If the data set contains 'm' tuples then each attribute of the data set has 'm' values. For each attribute 'm' intervals are generated and within each interval 'n' Uniform distribution data error corrected values are generated and the entropy is calculated for all these error corrected values and then one best split point is selected for each interval. One optimal split point is selected from all the best points of all the interval of one particular attribute. Same process is repeated for all attributes of the training data set. Finally, one optimal split attribute and optimal split point is selected from 'k' attributes and k(mn – 1) potential split points. Optimal split attribute and optimal split point constitutes optimal split pair. The Uniform decision tree for uncertain data (UDTUD) algorithm constructs a decision tree classifier splitting each node into left and right nodes. Initially, the root node contains all the training data tuples. A set of 'n' sample values are generated using uniform distribution model for each value of an attribute and for all attributes in the training data set and then stored in the root node. Entropy values are computed for k(mn – 1) split points where k is the number attributes of the training data set, m is the number of training data tuples at the current node T and 'n' is the number of uniform error corrected values for each attribute value in the training data set. The process of partitioning the training data tuples in a node into two subsets based on the best split point value z_T of best split attribute A_{j_T} and storing the resulting tuples in its left and right nodes is referred to as splitting.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

After splitting of the root node into two left and right sub-nodes the same process is applied for both left and right nodes. The recursive process stops when all the divided tuples have the same class or less than a threshold value specified at a particular node. Extensive experiments have been conducted which show that the resulting experiments are more accurate than those of certain decision trees (CDT). UDTUD can build more accurate decision tree classifiers but computational complexity of UDTUD is 'n' times expensive than CDT. To reduce the computational complexity of UDTUD we have proposed a pruning technique so that entropy is calculated only at one best point for each interval. Accuracy and execution time of UDTUD algorithm for 9 data sets are shown in Table 6.3 and comparison of execution time and accuracy for CDT and UDTUD algorithms for 9 data sets are shown in Table 6.4 and charted in Figure 6.1 and Figure 6.2 respectively.

5.2 Pseudo code for Uniform Decision Tree for Uncertain data (UDTUD) Algorithm

UNIFORM_UNCERTAIN_DECISION_TREE (T)

1. If all the training tuples in the node T have the
2. same class label then
3. set $p_T(c) = 1.0$
4. return(T)
5. If tuples in the node T have more than one class then
6. **For each value of each numerical attribute in the training data set construct an interval and then find entropy at 'n' Uniform distribution error corrected values in the interval and then select one optimal point, point with the minimum entropy, in the interval. If the training data set contains 'm' tuples then for each numerical attribute 'm' interval are generated and finally one best optimal split point is selected from n(m – 1) possible potential split points**
7. Find_Best_Split(T)
8. For i ← 1 to datasize[T] do
9. If split_atribute_value[t_i] <= split_point[T] then
10. Add tuple t_i to left[T]
11. Else
12. Add tuple t_i to right[T]
13. If left[T] = NIL or right[T] = NIL then
14. Create empirical probability distribution of the node T
15. return(T)
16. If left[T] != NIL and right[T] != NIL then
17. UNIFORM_UNCERTAIN_DECISION_TREE(left[T])
18. UNIFORM_UNCERTAIN_DECISION_TREE(right[T])
19. return(T)

6. EXPERIMENTAL RESULTS

A simulation model is developed for evaluating the performance of two algorithms: Certain Decision Tree (CDT) and Uniform Decision Tree (UDTUD) experimentally. The data sets shown in Table 5.1 from University of California (UCI) Machine Learning Repository are employed for evaluating the performance of the above said algorithms.

No	Data Set Name	Training Tuples	No. Of Attributes	No. Of Classes	Test Tuples
1	Iris	150	4	3	10-fold
2	Glass	214	9	6	10-fold
3	Ionosphere	351	32	2	10-fold
4	Breast	569	30	2	10-fold
5	Vehicle	846	18	4	10-fold
6	Segment	2310	14	7	10-fold
7	Satellite	4435	36	6	2000
8	Page	5473	10	5	10-fold
9	Pen Digits	7494	16	10	3498

Table 6.1 Data Sets from the UCI Machine Learning Repository

In all our experiments we have used data sets from the UCI Machine Learning Repository [6]. 10-fold cross-validation technique is used for test tuples for all training data sets with numerical attributes except Satellite and PenDigits training data sets [6]. For Satellite and PenDigits training data sets with numerical attributes a separate test data set is used for testing because sufficient test tuples are available.

The simulation model is implemented in Java 1.6 on a Personal Computer with 3.22 GHz Pentium Dual Core processor (CPU), and 2 GB of main memory (RAM). The performance measures, accuracy and execution time (in seconds), for the above said algorithms are presented in Table 6.2 to Table 6.4 and Figure 6.1 to Figure 6.2.

No	Data Set Name	Total Tuples	Accuracy	Execution Time
1	Iris	150	97.3333	1.1
2	Glass	214	89.6213	1.2
3	Ionosphere	351	83.1429	1.37
4	Breast	569	97.3214	2.462
5	Vehicle	846	79.0476	6.6
6	Segment	2310	96.5801	27.787
7	Satellite	4435	83.25	146.03
8	Page	5473	98.5612	34.26
9	Pen Digits	7494	90.9096	644.164

Table 6.2 Certain Decision Tree (CDT) Accuracy and Execution Time

No	Data Set Name	Total Tuples	Accuracy	Execution Time
1	Iris	150	98.0	1.1
2	Glass	214	94.29	1.2
3	Ionosphere	351	98.0312	17.361

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

4	Breast	569	97.9815	25.462
5	Vehicle	846	95.1081	34.432
6	Segment	2310	96.624	216.532
7	Satellite	4435	84.6791	296.5412
8	Page	5473	98.9365	339.5132
9	Pen Digits	7494	91.9632	916.2311

Table 6.3 Uniform Decision Tree (UDTUD) Accuracy and Execution Time

No	Data Set Name	CDT Accuracy	UDTUD Accuracy	CDT Execution Time	UDTUD Execution Time
1	Iris	97.3333	98.0	1.1	1.1
2	Glass	89.6213	94.29	1.2	1.2
3	Ionosphere	83.1429	98.0312	1.37	17.361
4	Breast	97.3214	97.9815	2.462	25.462
5	Vehicle	79.0476	95.1081	6.6	34.432
6	Segment	96.5801	96.624	27.787	216.532
7	Satellite	83.25	84.6791	146.03	296.5412
8	Page	98.5612	98.9365	34.26	339.5132
9	Pen Digits	90.9096	91.9632	644.164	916.2311

Table 6.4 Comparison of accuracy and execution times of CDT and UDTUD

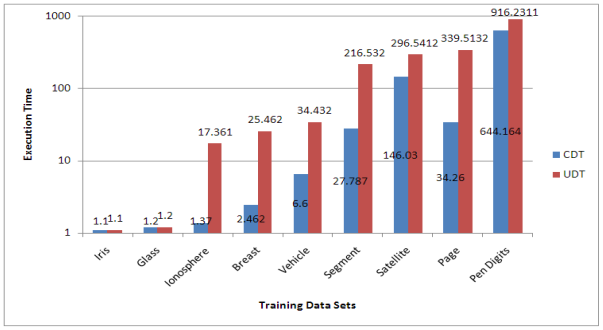

Figure 6.1 Comparison of execution times of CDT and UDTUD

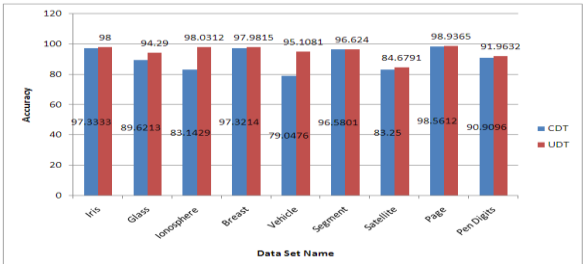

Figure6.2 Comparison of Classification Accuracies of CDT and UDTUD

7. CONCLUSIONS

7.1 Contributions

The performance of existing traditional or classical or certain decision tree (CDT) is verified experimentally through simulation. A new decision tree classifier construction algorithm called Uniform Decision Tree for Uncertain Data (UDTUD) is proposed and compared with the existing Certain Decision Tree classifier (CDT). It is found that the classification accuracy of proposed algorithm (UDTUD) is much better than CDT algorithm.

7.2. Limitations

Proposed algorithm, Uniform Decision Tree for Uncertain Data (UDTUD) classifier construction, handles only data uncertainty present in the values of numerical attributes of the training data sets only. Also computational complexity of UDTUD is very high and execution time of UDTUD is more for many of the training data sets.

7.3. Suggestions for future work

Special techniques or ideas or plans are needed to handle different types of data uncertainties present in the training data sets. Special methods are needed to handle data uncertainty in categorical attributes also. Special pruning techniques are needed to reduce execution time of UDTUD. Also special techniques are needed to find and correct random noise and other errors in the categorical attributes.

REFERENCES

[1] Jiawei Han, Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, second edition, 2006. pp.285–292
[2] Introduction to Machine Learning Ethem Alpaydin PHI MIT Press, second edition. pp. 185–188
[3] SMITH Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho, and Sau Dan Lee "Decision Trees for Uncertain Data" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.23, No.1, JANUARY 2011
[4] Hsiao-Wei Hu, Yen-Liang Chen, and Kwei Tang "A Dynamic Discretization Approach for Constructing Decision Trees with a Continuous Label" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL.21, No.11, NOVEMBER 2009
[5] R.E. Walpole and R.H. Myers, Probability and Statistics for Engineers and Scientists. Macmillan Publishing Company, 1993.
[6] A. Asuncion and D. Newman, UCI Machine Learning Repository, http://www.ics.uci.edu/mlearn/MLRepository.html, 2007.
[7] U.M. Fayyad and K.B. Irani, "On the Handling of Continuous –Valued Attributes in Decision tree Generation",Machine Learning, vol. 8, pp. 87-102, 1996.

BIOGRAPHIES

MR. C SUDARSANA REDDY

M.Tech. in Computer Science and Engineering (Gold Medalist) from Sri Venkateswara University College of Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India and also MCA from the same college and same university. More than 18 years of teaching experience in various engineering and P.G. colleges affiliated to Sri Venkateswara University and Jawaharlal Nehru technological University (JNTU), Anathapur, Andhra Pradesh, India.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

S. AQUTER BABU
Master of Computer Applications from Sri Venkateswara University, Tirupati. Assistant Professor (Sr. Scale), Dept. of Computer Science, Dravidian University, Kuppam, Pin code - 517 425. Andhra Pradesh, India. U.G.C. NET Qualified in Computer Applications Subject and Pursuing Ph.D. in Computer Science.

Dr. V. VASU
M.Sc, PhD in Mathematics from Sri Venkateswara University, Tirupati. Currently working as an academic consultant in department of Mathemetics, from Sri Venkateswara University since 2004.