Performance Evaluation of Decision Trees for Uncertain Data Mining

Ms. Kiran Dhandore[1], Dr. Lata Ragha[2]

[1]Terna Engineering College, Dept. Computer Engineering,
Nerul, Navi-Mumbai-400706

[2]Terna Engineering College, Dept. Computer Engineering,
Nerul, Navi-Mumbai-400706

Abstract

Decision trees are considered to be one of the most popular approaches for representing classifiers. Researchers from various disciplines such as statistics, machine learning, pattern recognisition and Data Mining have dealt with the issue of growing a decision tree from available data. Traditional decision tree classifiers used to work on the data whose values are known and precise. Therefore in this research such classifiers are used to handle data with uncertain information. Two approaches has been implemented one is the averaging in which the data uncertainty is represented by abstract probability distribution by summary statistics such as mean and variances. The other approach is the distribution based which considers all the sample points that constitute each pdf.

Keywords:- Decision Trees, Uncertain data, Pruning, Entropy, Classification.

1. INTRODUCTION

Data is often associated with uncertainty because of measurement inaccuracy, sampling discrepancy, outdated data sources, or other errors. In recent years, uncertain data has become ubiquitous because of new technologies for collecting data which can only measure and collect the data in an imprecise way. 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. Many scientific measurement techniques are inherently imprecise. In such cases, the level of uncertainty may be derived from the errors in the underlying instrumentation. Many new hardware technologies such as sensors generate data which is imprecise. In such cases, the error in the sensor network readings can be modeled, and the resulting data can be modeled as imprecise data. In many applications such as the tracking of mobile objects, the future trajectory of the objects is modeled by forecasting techniques. Small errors in current readings can get magnified over the forecast into the distant future of the trajectory. This is frequently encountered in cosmological applications when one models the probability of encounters with Near-Earth-Objects (NEOs). Errors in forecasting are also encountered in non-spatial applications such as electronic commerce. In many applications such as privacy-preserving data mining, the data is modified by adding perturbations to it. In such cases, the format of the output is exactly the same as that of uncertain data. Location-based services: in the scenario of moving objects (such as vehicles or people), it is impossible for the database to track the exact locations of all objects at all time instants. Therefore, the location of each object is associated with uncertainty between updates. In recent years, there has been much research on the management of uncertain data in databases, such as the representation of uncertainty in databases and querying data with uncertainty. However, little research work has addressed the issue of mining uncertain data. We know that with uncertainty, data values are no longer atomic. To apply traditional data mining techniques, uncertain data has to be summarized into atomic values.

2. RELATED WORKS

There has been a growing interest in uncertain data mining and recently more research has been conducted on that. Most of them focus on clustering uncertain data [1], [2]. 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. In [3], 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 for sets of samples vs. single values. To improve the performance of UK-means, pruning techniques have been proposed [4]. Xia et al. [5] introduce a new conceptual clustering algorithm for uncertain categorical data**.** Agarwal [6] proposes density based transforms for uncertain data mining. There is also some research on identifying frequent item sets and association mining from uncertain datasets [7]. The support of item sets and confidence of association rules are integrated with the existential probability of transactions and items. Burdicks [8] discuss OLAP computation on uncertain data. None of them address the issue of developing a general classification and prediction algorithm for uncertain data.

Decision trees are one of the most important aspects for “Decision-making”. Classification is one of the most widespread data mining problems found in real life. Decision tree classification is one of the best-known solution approaches [9], [10], [11]. In C4.5 and probabilistic decision trees, missing values in training data are handled by using fractional tuples [12]. During testing, each missing value is replaced by multiple values with probabilities based on the training tuples, thus allowing probabilistic classification results. They have adopted the technique of fractional tuple for splitting tuples into subsets when the domain of its pdf spans across the split point.

3. EXISTING SYSTEM

In traditional decision-tree classification, a feature or an attribute of a tuple is either categorical or numerical. For the latter, a precise and definite point value is usually assumed. In many applications, however, data uncertainty is common. 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. Although the previous techniques can improve the efficiency of means, they do not consider the spatial relationship among cluster representatives, nor make use of the proximity between groups of uncertain objects to perform pruning in batch. A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances called averaging approach. Another approach is to consider the complete information carried by the probability distributions to build a decision tree called Distribution-based approach.

4. PROPOSED SYSTEM

The main goal of the decision tree is to construct by using:

(1) The basic algorithm to construct decision trees out of uncertain datasets.

(2) Find out whether the Distribution-based approach could lead to higher classification accuracy compared with the Averaging approach.

(3) Establish a theoretical foundation on which pruning techniques are derived that can significantly improve the computational efficiency of the Distribution-based algorithms.

The following figure 2 shows the flow diagram of the proposed methodology.

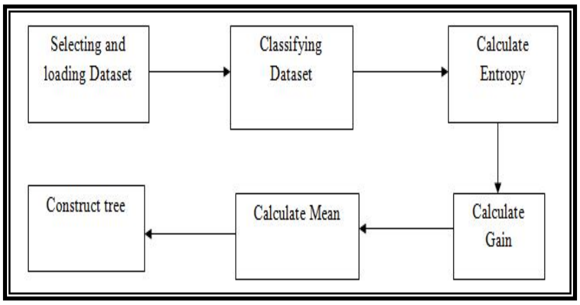

Figure 1: Proposed Methodology

The two approaches for handling uncertain data. The first approach, called "Averaging", transforms an uncertain dataset to a point-valued one by replacing each pdf with its mean value. To exploit the full information carried by the pdf's, the second approach, called "Distribution-based", considers all the sample points that constitute each pdf.

A. Averaging

A simple way to handle data uncertainty is to abstract probability distributions by summary statistics such as means and variances which is called as the averaging approach. A straightforward way to deal with the uncertain information is to replace each pdf with its expected value, thus effectively converting the data tuples to point-valued tuples. The algorithm starts with the root node and with S being the set of all training tuples. At each node n, we first check if all the tuples in S have the same class label.

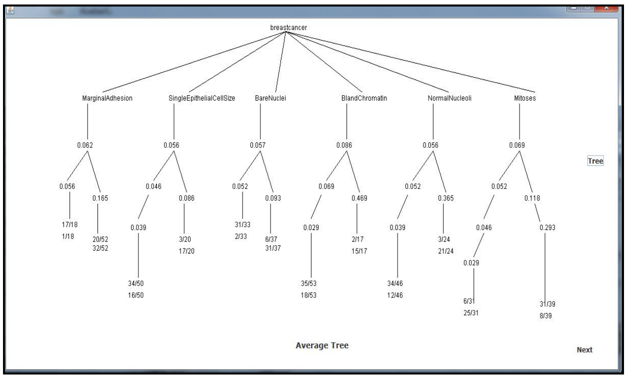

Figure 2: Averaging tree

B. Distribution Based

An approach is to consider the complete information carried by the probability distributions to build a decision tree which is called as Distribution-based approach. After an attribute Ajn and a split point zn has been chosen for a node n, we split the set of tuples S into two subsets L and R. The major difference from the point-data case lies in the way the set S is split. If the pdf properly contains the split point, i.e., ai,jn ≤ zn < bi,jn, we split ti into two fractional tuples[3] tL and tR and add them to L and R, respectively. The algorithm is called UDT (Uncertain Decision Tree).

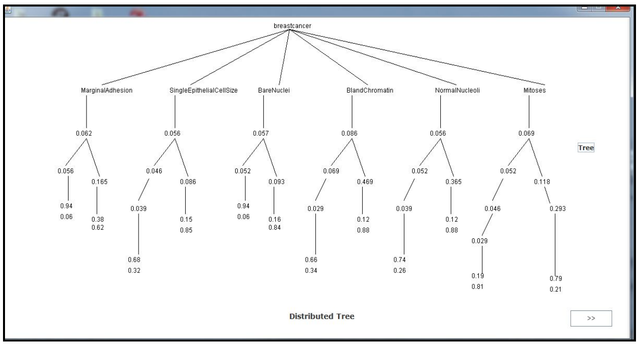

Figure 3: Distributed Tree

5. PRUNING TECHNIQUES

Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. The dual goal of pruning is reduced complexity of the final classifier as well as better predictive accuracy by the reduction of over fitting and removal of sections of a classifier that may be based on noisy or erroneous data. Over fitting is a significant practical difficulty for decision tree models and many other predictive models.

Over fitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an increased test set error. There are several approaches to avoiding over fitting in building decision trees. Pre-pruning that stop growing the tree earlier, before it perfectly classifies the training set. Post-pruning that allows the tree to perfectly classify the training set, and then post prune the tree. Although UDT can build a more accurate decision tree, it is not as efficient as AVG. To determine the best attribute and split point for a node, UDT has to examine k (ms - 1) split points, where k = number of attributes, m = number of tuples, and s = number of samples per pdf. (AVG has to examine only k (m - 1) split points.) For each such candidate attribute Aj and split point z, an entropy H (z;Aj) has to be computed . Entropy calculations are the most computation-intensive part of UDT. Our approach to developing more efficient algorithms is to come up with strategies for pruning candidate split points and entropy calculations.

5.1 PRUNING BY BOUNDING (UDT-BP)

Algorithm UDT-BP has to examine all end-points of empty and homogeneous intervals as well as all sample points in heterogeneous intervals. First we compute the entropy H(q;Aj) for all end-points q belongs to Qj . Let H* j = minq2QjfH (q; Aj) g be the smallest of such end-point entropy values. Next, for each heterogeneous interval (a; b], we compute a lower bound, Lj , of H(z;Aj) over all candidate split points z belongs to (a; b]. If Lj H*j , we know that none of the candidate split points within the interval (a; b] can give an entropy that is smaller than H* j and thus the whole interval can be pruned. We note that the number of end-points is much smaller than the total number of candidate split points. So, if a lot of heterogeneous intervals are pruned in this manner, we can eliminate many entropy calculations. So, the key to this pruning technique is to find a lower bound of H (z; Aj) that is not costly to compute, and yet is reasonably tight for the pruning to be effective.

5.2 END POINT SAMPLING (UDT-ES)

In some settings, UDT-GP reduces the number of ―entropy calculations (including the calculation of entropy values of split points and the calculation of entropy like lower bounds for intervals) to only 2.7% of that of UDT. On a closer inspection, we find that many of these remaining entropy calculations come from the determination of end-point entropy values. In order to further improve the algorithm’s performance, we propose a method to prune these end-points. We note that the entropy H (q;Aj) of an end-point q is computed for two reasons. Firstly, for empty and homogeneous intervals, their end-points are the only candidates for the optimal split point. Secondly, the minimum of all end-point entropy values is used as a pruning threshold. For the latter purpose, we remark that it is unnecessary that we consider all end-point entropy values. We can take a sample of the end-points (say 10%) and use their entropy values to derive a pruning threshold. This threshold might be slightly less effective as the one derived from all end-points; however, finding it requires much fewer entropy calculations. Also, we can concatenate a few consecutive intervals, say I1; I2; I3, into a bigger interval I, compute a lower bound and attempt to prune. If successful, we have effectively pruned the end-points of I1, I2 and I3. We incorporate these End-point Sampling intervals obtained after pruning is Y00 (row 9), which is a much smaller candidate than the set of candidate intervals when no end-point sampling is used. For the candidate intervals in Y00, we compute the values H(z;Aj) for all pdf sample points to find the minimum entropy value.

6. RESULTS AND DISCUSSION

The accuracy of a decision tree classifier can be much improved if the “complete information” of a data item (taking into account the probability density function (pdf)) is utilised. Distribution based algorithm can improve classification accuracy because there are more choices of split points. The distribution approach has to examine k (ms-1) split points whereas the AVG approach has to examine k(m-1) split points. Entropy calculations are the most computation intensive part of UDT. To explore the potential of achieving a higher classification accuracy by considering data uncertainty, we have implemented AVG and UDT and applied them to 04 datasets namely glass dataset, page block , Japanese Vowel, Breast Cancer dataset taken from the UCI Machine Learning Repository. These datasets are chosen because they contain mostly numerical attributes obtained from measurements. We model uncertainty information by fitting appropriate error models on to the point data. For each tuple ti and for each attribute Aj , the point value vi;j reported in a dataset is used as the mean of a pdf fi;j , defined over an interval [ai;j ; bi;j]. The range of values for Aj (over the whole data set) is noted and the width of [ai;j ; bi;j] is set to w _ jAj j, where jAj j denotes the width of the range for Aj and w is a controlled parameter. To generate the pdf fi;j , we consider two options. The first is uniform distribution, which implies fi;j(x) = (bi;j - ai;j)-1. The other option is Gaussian distribution, for which we use 1/4 (bi;j -ai;j) as the standard deviation. In both cases, the pdf is generated using s sample points in the interval. Using this method (with controllable parameters w and s, and a choice of Gaussian vs. uniform distribution), we transform a data set with point values into one with uncertainty. The reason that we choose

Gaussian distribution and uniform distribution is that most physical measures involve random noise which follows Gaussian distribution, and that digitisation of the measured values introduces quantisation noise that is best described by a uniform distribution.

TABLE 1: Accuracy Improvement by Considering the Distribution

DATASET	AVG	BEST CASE	UDT (Gaussian Distribution)
Japanese Vowel	81.89	87.30	87.30
Page Block	95.73	96.82	96.82
Glass	66.49	72.75	69.60
Breast Cancer	93.52	95.93	94.73

From the table, we see that UDT builds more accurate decision trees than AVG does for different distributions over a wide range of w. For the first data set, whose pdf is modelled from the raw data samples, the accuracy is improved from 81.89% to 87.30%. Also the following accuracy classification graph shows UDT gives better accuracy than averaging.

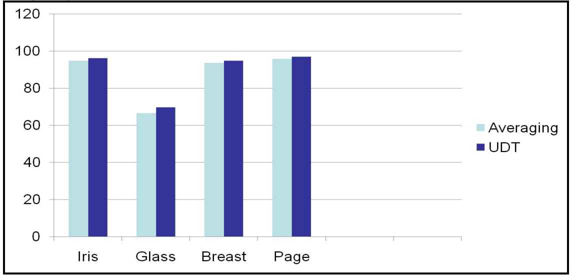

Figure 4: Graph of Accuracy classification

7. PERFORMANCE EVALUATION

The performance of the decision trees can be done by implementing pruning techniques. Two techniques has been implemented one is UDT-BP pruning technique and the other is UDT-ES pruning technique. Based on the parameters graphs with respect to time and pruning techniques along with the dataset are plotted. For pruning effectiveness the graph is with respect to number of entropy calculations versus pruning techniques.

A. Execution Time

The efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations (space complexity). The following graph shows the pruning techniques UDT-BP and UDT-ES are plotted on X-axis w.r.t time in milliseconds on Y-axis. Thus UDT-ES takes less time and efficient in building the decision tree so that we achieve better classification accuracy. AVG constructs different decision from UDT-based algorithms, and that AVG generally constructs algorithms which have low accuracy. The AVG algorithm does not show any information about uncertainty, takes very less time to finish. But it is not as accurate as the distribution-based algorithms.

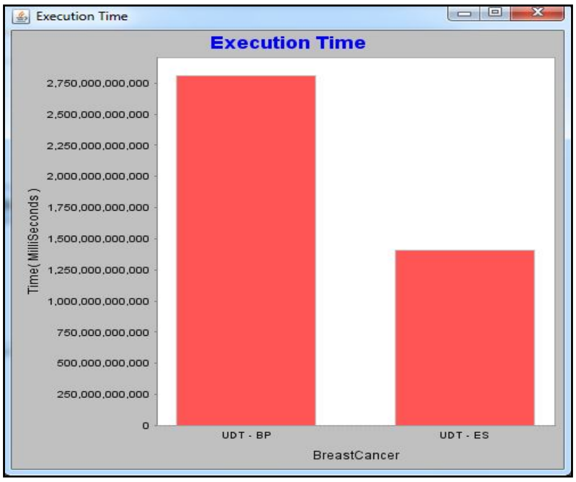

Figure 5: Execution Time

B. Pruning Effectiveness

Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. The dual goal of pruning is reduced complexity of the final classifier as well as better predictive accuracy by the reduction of over fitting and removal of sections of a classifier that may be based on noisy or erroneous data. Since the computation time of the lower bound of an interval is comparable to that of computing entropy, the number of entropy calculations for UDT-BP, UDT-ES include the number of lower bounds computed. Figure 6 shows the number of entropy calculations performed by the algorithm. The figure shows that our pruning techniques are highly effective. Indeed, UDT-ES reduces the number of entropy calculations when compared with UDT-BP. As entropy calculations dominate the execution time of UDT, such effective pruning techniques significantly reduce the tree-construction time

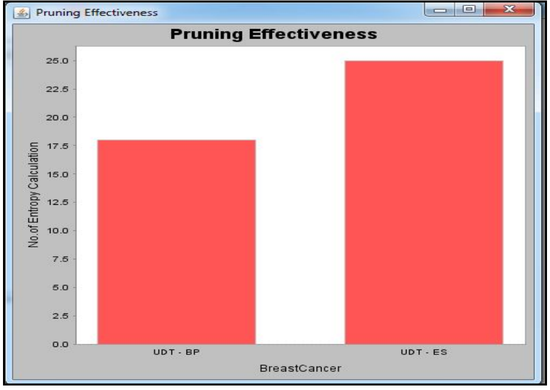

Figure 6: Pruning Effectiveness

8. CONCLUSION

The model of decision-tree classification has been extended to accommodate data tuples having numerical attributes with uncertainty described by arbitrary pdf's. We have modified classical decision tree building algorithms to build decision trees for classifying such data. It has been found empirically that when suitable pdf’s are used, exploiting data uncertainty leads to decision trees with remarkably higher accuracies. Performance is an issue, though, because of the increased amount of information to be processed, as well as the more complicated entropy computations involved. Therefore, we have devised a series of pruning techniques to improve tree construction efficiency. Their execution times are of an order of magnitude comparable to classical algorithms. Some of these pruning techniques are generalizations of analogous techniques for handling point-valued data. Other techniques, namely pruning by bounding and end-point sampling are novel. Although our novel techniques are primarily designed to handle uncertain data, they are also useful for building decision trees using classical algorithms when there are tremendous amounts of data tuples.

ACKNOWLEDGEMENT

I express my sincere thanks to my Project Guide Dr. Lata Ragha HOD of Computer Department for her excellent guidance and supervision, assisting with all kinds of support and inspiration, and valuable suggestions throughout this investigation and preparation of this project.

REFERENCES

[1] Michael Chau, Reynold Cheng, and Ben Kao “Uncertain Data Mining: A New Research Direction”, published in Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge discovery and Data Mining.

[2] Smith Tsangy ,Ben Kao, Kevin Y. Yip, Wai-Shing Ho, Sau Dan Lee “Decision Trees for Uncertain Data” Knowledge and Data Engineering, IEEE Transactions on Volume 23, Issue 1,Jan 2011 pp.64-78

[3] Varsha Choudhary, Pranita Jain, “Classification: A Decision Tree for Uncertain Data Using CDF” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 3, Issue 1, January -February 2013, pp.1501-1506

[4] W. Street, W. Wolberg, and O. Mangasarian, “Nuclear feature extraction for breast tumor diagnosis,” in SPIE, vol. 1905, San Jose, CA, U.S.A., 1993, pp. 861–870. [Online]. Available: http: //citeseer.ist.psu.edu/street93nuclear.html

[5] 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.

[6] J.R, QUINLAN ‘Induction of Decision Trees” Machine Learning 1: 81-106.

[7] R.Agrawal, T.Imielinski and A.N Swami. Database Mining: A performance perspective. IEEE Trans. Knowl. Data Eng., vol.5, no.6, pp. 914-925, 1993

[8] Y. Sahin and E. Duman “Detecting Credit Card Fraud by Decision Trees and Support Vector Machines” Proceedings of the International MultiConference of Engineers and Computer Scientists 2011 Vol-I IMECS 2011.

[9] Hawarah L, Simonet A, Simonet M “Dealing with Missing Values in a Probabilistic Decision Tree during Classification”, The Second Intern8ational Workshop on Mining Complex Data, pp. 325-329.

[10] Data Mining Analysis (breast-cancer data) Jung-Ying Wang Register number: D9115007, May, 2003.

[11] Khumesh Patil, Namrata Pagare, Pallavi Narkhede, Prashant Brahmankar “ Classifying Climate Data (uncertain) using Decision Tree” International Journal of Advanced Research (2014), Volume 2, Issue 4, 402-408 ,ISSN 2320-5407.

[12] Nikita Patel , Saurabh Upadhyay” Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA” International Journal of Computer Applications (0975 – 8887) Volume 60– No.12, December 2012