Performance Analysis on Uncertain Data using Decision Tree

Bhosale J.D.
M.B.E.Society's College of Engineering, Ambajogai, Maharashtra

Patil B.M.
M.B.E.Society's College of Engineering, Ambajogai, Maharashtra

ABSTRACT

Data uncertainty is common in emerging applications, such as sensor networks, moving object databases, medical and biological fields. Data uncertainty can be caused by various factors including measurements precision limitation. Data uncertainty is inherited in various applications due to different reasons such as outdated sources or imprecise measurement and transmission problems. Classification is one of the most popular data mining techniques. Lot of people used decision tree for data classification and it widely used on certain or precise data. However in this paper we applied on uncertain data which is taken from UCI machine learning repository. This paper proposes a decision tree based classification method on uncertain data. We construct decision tree algorithms by including entropy and information gain, considering the uncertain data intervals. We use some pruning techniques that can improve efficiency of the decision tree and our experiment show that it significantly reduce the tree-construction time.

Keywords

Uncertain Data, Decision Tree, Classification, Data Minings

1. INTRODUCTION

Now a day, huge amount of historical information handling is very important. The data mining is very effective tool for extracting knowledge from historical information. There are three data mining techniques the Naive Bayes, the neural network and the decision tree algorithms and concluded that C4.5 decision tree algorithm has a much better performance than the other two techniques based on their researches [8]. Decision tree is a simple and popular technique in classification model so it is widely used method for classification modelling. They are easy to understand and practical. Decision tree algorithms begin with a set of cases, or examples, and create a tree data structure that can be used to classify new cases. Each case is described by a set of attributes or features which can have numeric or symbolic values. Associated with each training case is a label representing the name of a class. Each internal node of a decision tree contains a test, the result of which is used to decide what branch to follow from that node [3]. The partitioning process terminates when the subsets cannot be partitioned any further using predefined criteria.

ID3 [2] and C4.5 [3] are the algorithms used to construct decision tree. Decision trees are used in many applications. For example, image recognition, medical diagnosis [4], credit rating of loan applicants, scientific tests, fraud detection, target marketing, 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. Uncertainty arises due to number of factors, such as unreliable data transmission and data staling, the random nature of the physical data generation and collection process, measurement and decision errors. For example, there are massive amounts of uncertain data in sensor networks, such as temperature, humidity, and pressure. [5]Since data uncertainty is important to develop classification models for uncertain data. For instance, a tumor is typically classified as benign (2) or malignant (4) in cancer diagnosis and treatment. 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. In this paper we studies decision tree based classification methods for uncertain data using *Averaging* and *Distribution* method. In this paper, our focus on the decision tree based classification approach. We choose the decision tree because of it have number of positive features. Decision tree is simple to understand and implement. It requires little data preparation compare to other methods.

Our goals are as follows:

1. Develop an algorithm for building decision trees from uncertain data using the Averaging-based approach;
2. Build decision tree using Distribution-based approach.
3. We prove through experiments that pruning satisfactory performance even when the training data is highly uncertain.
4. Pruning improves greater efficiency of construction of decision tree.

This paper is organized as follows. In the next section, we will discuss related work. Section 3 describes the different algorithm in detail. Section 4 shows that how pruning effective to improve execution time to built decision tree. Section 5 illustrates experimental results in detail. The Section 6 concludes the paper.

2. RELATED WORK

Classification is an important area in data mining. Many classification algorithms have been given in the literature, such as decision tree classifiers [3], Bayesian classifiers [7] and artificial neural networks [3].Decision tree classification with missing data has been addressed for decades in the form of missing values [12].

In C4.5 [3], is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is a software extension and thus improvement of the basic ID3 algorithm designed by Quinlan. The decision trees generated by C4.5 can be used for classification, and for this reason, C4.5 is often referred to as a statistical classifier. For inducing classification rules in the form of Decision Trees from a set of given examples C4.5 algorithm was introduced by Quinlan. C4.5 is a redevelopment of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on.

A decision tree construction on the data tuples with numerical is computationally demanding in nowadays [9]. Finding the best split point is computationally expensive. To improve efficiency, many techniques have been introduced to reduce the number of candidate split points [10], [9], [11]. Our work can be considered a pruning technique to reduce the split point.

3. ALGORITHMS

Follows are two proposed algorithms as follows for handling data uncertainty:

3.1 Averaging method

Averaging method (AVG) is used to transforms an uncertain data set into a point-valued one by replacing each pdf with its mean value. It converts data tuples into the point values. This method of construction of decision tree uses the C4.5 algorithm. We use C4.5 algorithm based on the entropy.

A greedy search strategy is used by this algorithm; the best attribute is picked by it and never reconsiders earlier choices by looking back. And this algorithm builds a tree top-down .When the algorithm start processing a node, we examine a set of tuples S. 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 c then we make n as a leaf node and probability is P_n (c)=1, P_n(c')=0.Otherwise, we select an attribute A_{jn} and a split point z_n and divide the tuples into two subsets "left" and "right." If $A_{jn} <= Z_n$ it is put into left branch of a tree. If $A_{jn} >= Z_n$ put it into the right branch. All tuples are put in the "left" subset L; and remaining to the "right" subset R. If R or L is empty then we cannot divide the tuples in S further. In such situation make n node as leaf node. Selection of A_{jn} and z_n is very important for constructing a good decision tree. The algorithm takes a set of tuples as a parameter to the algorithm and gives the best attribute and split point to those tuples.

3.2 Distribution method

Distribution based approach (UDT) is another method of tree constriction. The tree starts as a single node representing the training samples as a root node n. compute the entropy for root node. If entropy is zero, the samples are all of the same class then the node becomes a leaf and is labelled with that class. Otherwise, the algorithm calculates entropy. If entropy is non zero find the attribute A_j with highest information gain, as the criteria for selecting the attribute that will best separate the samples into an individual class. This attribute becomes the test attribute at the node.

The tree is split into two span as left and right according to test condition. The main difference is the point data lies in the training set S. Tuples t belongs to S i.e. training set t attribute Aj spans the interval [a, b] ,if b < = Z_n than pdf of t lies on left side of split point. So t is assigned to L. Likewise, we assign t to R if Z_n < = a. This algorithm recursively applies the same process to generate a decision tree for the given training samples.

The recursive process will stop only when one of the following conditions becomes true:

1) All samples in S for a given node belong to the same class
2) The samples may not be further divided if there are no remaining attributes.

4. PRUNING

In this section of paper pruning is described. At the time of tree building one problem arise is the optimal size of the final tree. When a decision tree is built, many of the branches will behave abnormally in the training data because of noise. Tree pruning methods address this problem of over fitting the data. However, it is hard to tell when a tree algorithm should stop because it is impossible to tell if the addition of a single extra node will dramatically decrease error. This problem is known as the horizon effect. A common strategy is to grow the tree until each node contains a small number of instances then use pruning to remove nodes that do not provide additional information. Pruning should reduce the size of a learning tree. There are many techniques for tree pruning that differ in the measurement that is used to optimize performance. Pruned trees are smaller and less complex and, thus, easier to understand.

An unpruned tree and a tree after pruning are shown in Figure 2 and Figure 5 respectively.There are two common approaches to tree pruning: pre-pruning and post-pruning.

In [13] the pre-pruning approach, a tree is "pruned" by stopping its construction early. It means by deciding not to further split the subset of training tuples at a given node. The second and more common approach is post-pruning. In this approach a subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most common class among the subtree being replaced.

The Distribution Based approach (UDT) can construct more accurate decision tree than Averaging method. Our approach to developing more efficient algorithms is to come up with strategies for pruning candidate split points and entropy calculations. To decide 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_n, entropy has to be calculated. Entropy calculations are the most computation part of UDT.

We are only pruning away candidate split points that give suboptimal entropy values. So, even after pruning, we are still finding optimal split points. Therefore, the pruning algorithms do not affect the resulting decision tree. It only eliminates suboptimal candidates from consideration, thereby speeding up the tree building process.

4.1 Basic Pruning algorithm

Algorithm UDT-BP has to examine all end points of empty and homogeneous intervals as well as all sample points in heterogeneous intervals. An interval is empty if no pdf's domain intersects it; an interval is homogeneous if all the pdf's that intersect it come from tuples of the same class. An interval is heterogeneous if it is neither empty nor homogeneous.

In order to further improve the algorithm's performance, we propose a method to prune these end-points. We can take a sample of the end-points and use their entropy values. This is UDT-ES as shown in fig.

5. EXPERIMENTS

In this section, we present the experimental results of the proposed decision tree algorithms. 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 Core 2 Duo 2.66GHz CPU. Our focus is on the pruning effectiveness of our pruning algorithms and their runtime performance.

The data sets selected from the UCI Machine Learning Repository [1]. These data sets are chosen because they contain mostly numerical attributes obtained from measurements. The Breast Cancer data set is used in this experiment. The UCI dataset has been modified accordingly our use. In this dataset the number of tuples are 70 and number of attributes are 11 out of which 10 are continuous and 1 is class label.

We first examine the execution time of the algorithms, which is charted in Figure 3. In this figure, bar is drawn for data set. The vertical axis, which is represents the execution time in seconds. We have given also the execution time of decision trees from those constructed by the UDT-based algorithms.

From the figure, we observe the following general (ascending) order of efficiency: UDT-BP, UDT-ES.

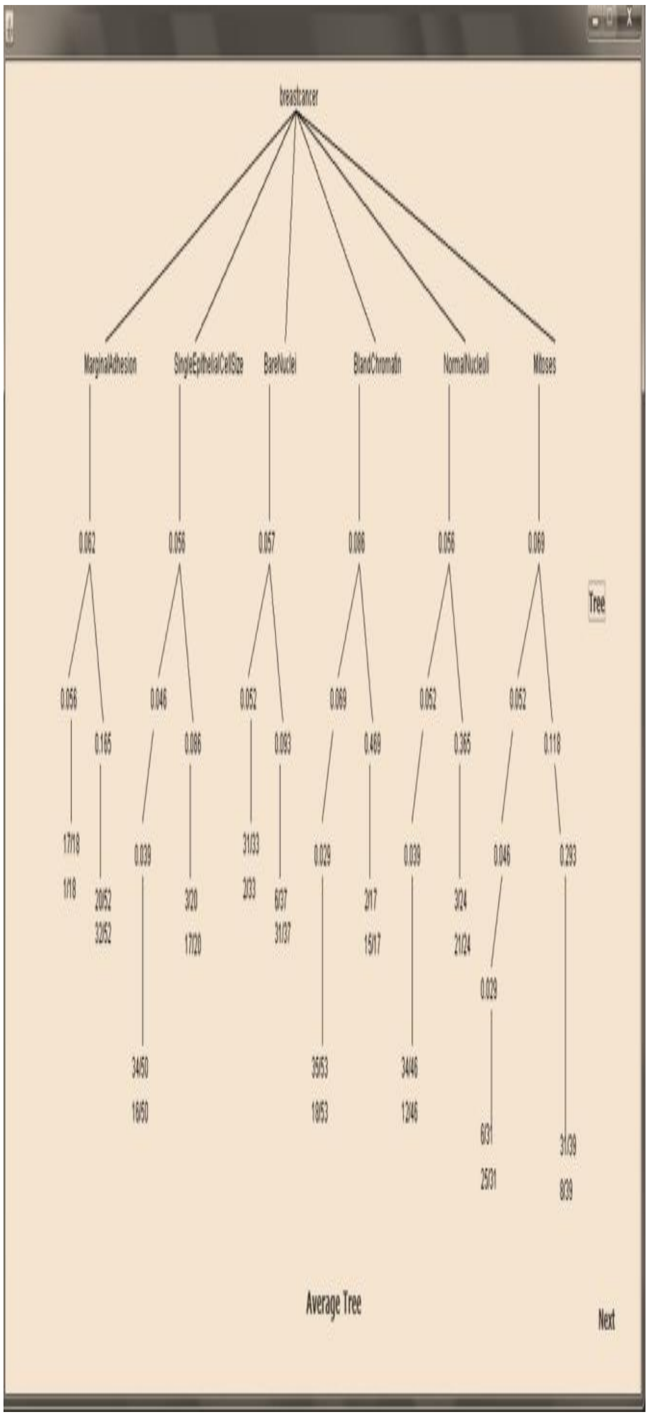

Fig 1: Decision Tree of Averaging Method

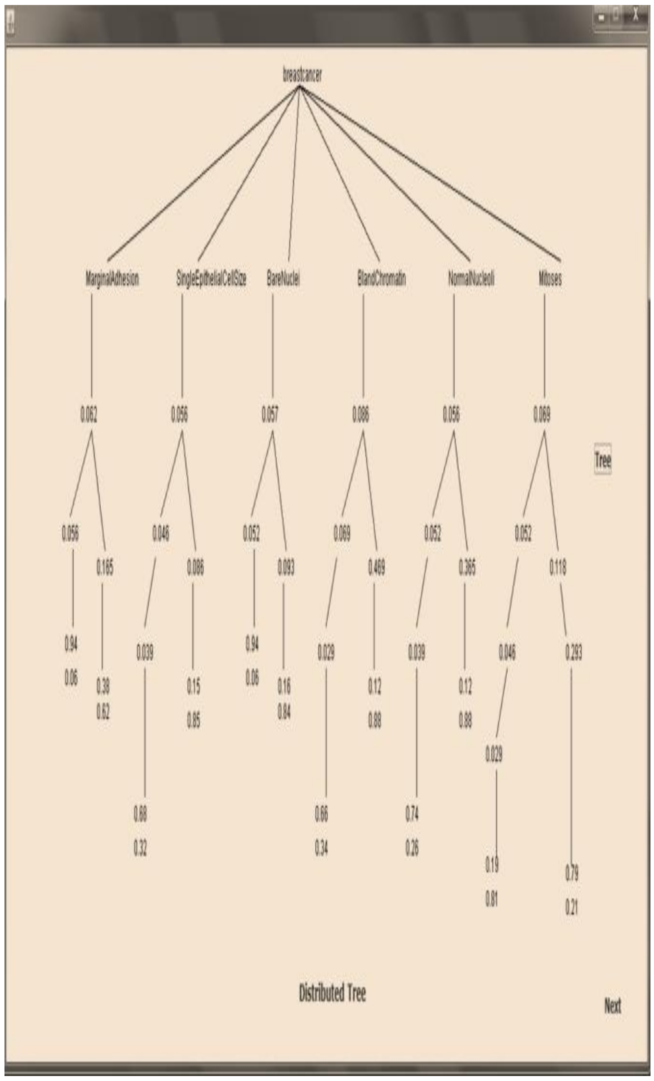

Fig 2: Decision Tree of Distribution Method

Next, we study the pruning effectiveness of the algorithms. Figure 4 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.

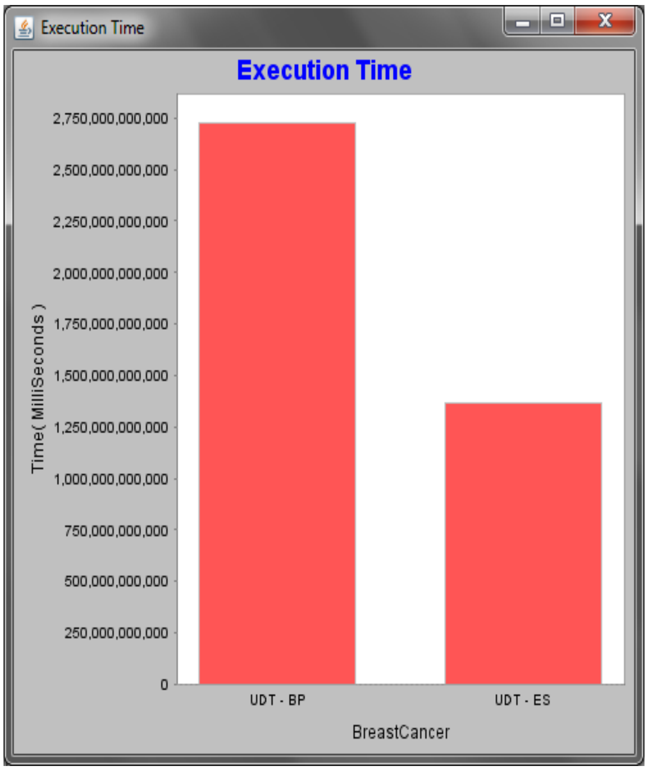

Fig 3: Execution time

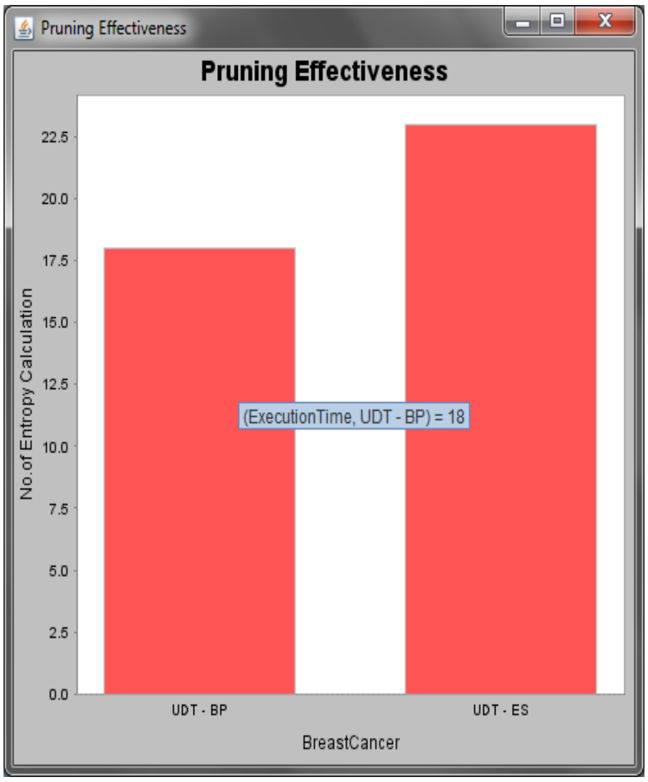

Fig 4: Pruning Effectiveness

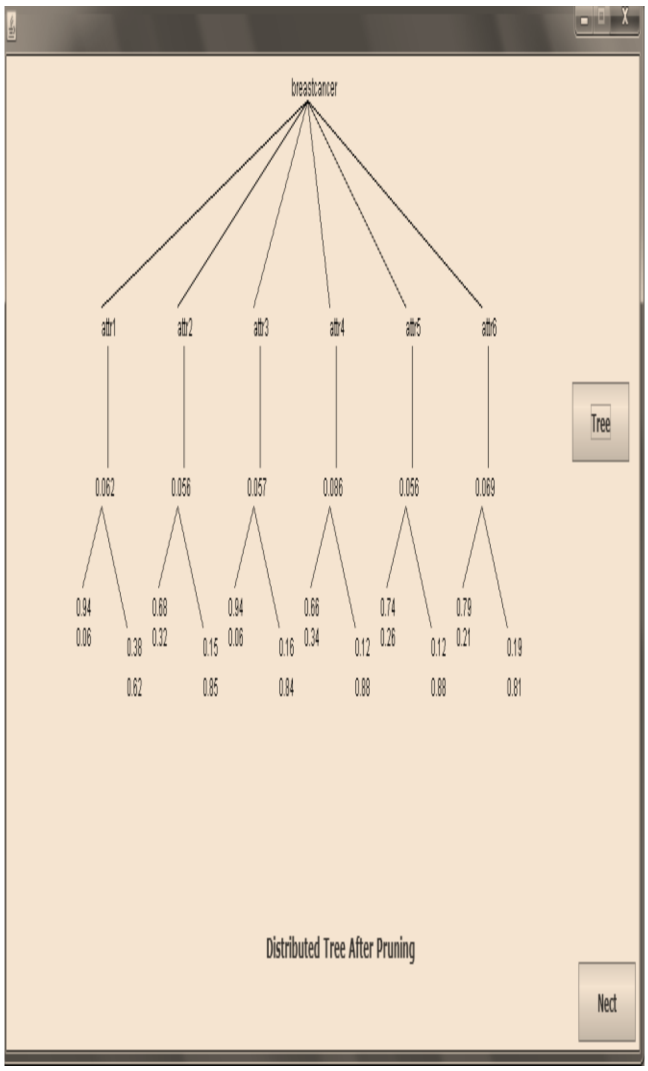

Fig 5: Tree after pruning

6. CONCLUSION

In this paper, we propose two decision tree algorithms. These algorithms are based on C4.5 algorithm. We calculate the measures used in decision tree, such as information entropy and information gain.

We have devised a series of pruning techniques to improve tree construction efficiency. Our algorithms have been experimentally verified to be highly effective. We conclude that execution time is improved because of pruning.

7. REFERENCES

[1] http://archive.ics.uci.edu/ml/datasets.html.

[2] J.R. Quinlan, "Induction of Decision Trees," Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[3] J.R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.

[4] C.L. Tsien, I.S. Kohane, and N. McIntosh, "Multiple Signal Integration by Decision Tree Induction to Detect Artefacts in the Neonatal Intensive Care Unit," Artificial Intelligence in Medicine, vol. 19, no. 3, pp. 189-202, 2000.

[5] W. Street, W. Wolberg, and O. Mangasarian, "Nuclear Feature "Extraction for Breast Tumor Diagnosis," Proc. SPIE, pp. 861-870, http://citeseer.ist.psu.edu/street93nuclear.html, 1993.

[6] L. Breiman, "Technical Note: Some Properties of Splitting Criteria," Machine Learning, vol. 24, no. 1, pp. 41-47, 1996.

[7] Langley P, Iba W, Thompson K (1992) An analysis of Bayesian classifiers, In: Proceedings of the tenth National Conference on artificial intelligence, pp. 223-228.

[8] Abdelghani Bellaachia, Erhan Guven, "Predicting Breast Cancer Survivability Using Data Mining Techniques", www.siam.org/meetings/sdm06/workproceed/bellaachia.pdf on Dec 06, 2010.

[9] T. Elomaa and J. Rousu, "General and efficient multisplitting of numerical Attributes", Machine learning, vol 36, no 3, pp 201-244, 1999.

[10] U. M. Fayyad and K. B. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Machine Learning*, 1992.

[11] T. Elomaa and J. Rousu, "Efficient multisplitting revisited: elimination of partition candidates," *Data Mining and knowledge Discovery*, vol. 8, no. 2, pp. 97–126, 2004.

[12] Hawarah L, Simonet A, Simonet M(2006) Dealing with Missing Values in a Probabilistic Decision Tree during Classification, The Second International Workshop on Mining Complex Data, pp. 325-329.

[13] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.