ISSN:0975-9646

Uncertain Data Mining using Decision Tree and Bagging Technique

Manasi M. Phadatare[#1], Sushma S. Nandgaonkar[*2]

[#1]*M.E. IInd year, Department of Computer Engineering,*
VP's College of Engineering, Baramati, India.
[*2]*Assistant Professor, Department of Computer Engineering,*
VP's College of Engineering, Baramati, India.

Abstract— **Classification is one of the important data mining techniques and Decision Tree is a most common structure for classification which is used in many applications. Decision tree classifier works on precise and known data. Traditional classifier extended to handle uncertain data caused by faulty data collection processes. To handle uncertainty feature value is represented by probability distribution function instead of single value. This improves accuracy of decision tree as complete information is used. Probability density function (PDF) requires many calculations. Pruning techniques are used to remove unwanted intervals and to reduce execution time. In this paper bagging method is combined with decision tree technique to stabilize the performance of decision tree and to improve accuracy of decision tree.**

Keywords— **Bagging; Classification; Decision Tree; Probability distribution; Uncertainty**.

I. Introduction

Data mining process employs one or more computer learning techniques to automatically analyze and extract knowledge from data in large databases Knowledge gained from data mining is in the form of model or generalization of data. Data mining techniques are broadly categorized in Supervised and Unsupervised data mining. Classification is probably best understood and widely used supervised data mining strategy. Classification techniques such as probabilistic summaries, decision trees, algebraic function, and support vector machine, etc. are used in various data mining applications. Classification became a successful data mining technique for certain data. In many applications data is often associated with uncertainty due to measurement inaccuracy, sampling discrepancy, outdated data sources, or other errors, there is need to develop classification and prediction technique for uncertain data. The main areas of research in this field are Modeling of uncertain data, Uncertain data management and, uncertain data mining. This paper focuses on uncertain data mining.

Many data mining applications are affected by the underlying uncertainty in the data. The data points may correspond to vaguely specified objects, and therefore considered uncertain in their representation. It is critical to design data mining techniques that can take such uncertainty into account during the computations. For instance, a tumor is typically classified as benign or malignant in cancer diagnosis and treatment. In practice, it is often very difficult to accurately classify a tumor due to the experiment precision limitation. Since data uncertainty is ubiquitous, it is important to develop classification models for uncertain data.

Decision Trees are a simple method for classification and predictive modeling. Decision tree can handle both categorical and numerical data. A decision tree is built by using a subset from instances of dataset and remaining subset test the accuracy of constructed tree. A decision tree partitions data into smaller segments called terminal nodes. Precise and definite point value is used to classify data tuple. Each terminal node is assigned a class label. The intermediate nodes, which include the root and other internal nodes, contain attribute test conditions to separate records that have different characteristics. The partitioning process terminates when the subsets cannot be partitioned further using predefined criteria. Many algorithms such as ID3, C4.5, Random tree, CART etc. have been devised for decision tree construction.

A. System Overview

The idea here is to enhance decision tree classifier to classify uncertain data accurately. Data set with uncertain values is input to system. Uncertain values include missing, repeated, stale, random values occurred due to errors in data collection process. We extend decision tree classifier to accurately classify tuples with uncertain values. It involves finding a good testing attribute and a good split point for each internal node, as well as an appropriate probability distribution over class label for each leaf node. In first step we calculate Gaussian probability distribution function for each attribute using sample points for good split point. In second step we create sequence of classifiers using bagging technique. For each classifier first we calculate information gain to select good attribute for classification. Then we use PDF for selected attribute to classify instances. Finally we test accuracy of constructed decision tree for uncertain data.

In this paper we consider missing values as uncertain data. We consider both the averaging and distribution based approach. In averaging approach we calculate average (mean) of PDF values of particular attribute. In distribution based approach we consider range of PDF values for particular attribute.

This paper is first to study classification of uncertain data using decision tree. Our contribution is to combine bagging method with existing decision tree to improve accuracy of decision tree. We used Gaussian PDF in closed interval to calculate probability distribution of a tuple over individual class label. We used decision tree algorithms like ID3, C4.5, and Random tree for decision tree construction and evaluated accuracy of decision tree against uncertain data.

This paper is organized as follows. In the next section, we will discuss previous work. Section III describes the proposed system for uncertain data model. It includes problem definition and solving approaches. Section IV gives results of experiments and important discussions. Finally section V concludes the paper.

II. LITERATURE SURVEY

A. *Fuzzy decision tree[7]*

Fuzzy information models data uncertainty arising from human perception and understanding. The uncertainty here is the vagueness and ambiguity of concepts, e.g. If temperature readings are considered then it is difficult to understand how hot is hot when available data value is "hot". In fuzzy classification, attributes as well as class labels can be fuzzy and are represented in fuzzy terms. In these models, a node of the decision tree does not give a crisp test that decides deterministically which branch down the tree training or testing tuple is sent.

B. *Classification based on Missing Values [8]*

Decision tree classification on uncertain data has been addressed for decades in the form of missing values. Missing values appear when some attribute values are not available during data collection or 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 (ordered attribute tree and probabilistic attribute tree).

C. *Existential and Value Uncertainty[2][3][8]*

Data uncertainty has been broadly classified into existential uncertainty and value uncertainty. Existential uncertainty appears when it is uncertain whether an object or a data tuple exists. Value uncertainty, on the other hand, appears when a tuple exists, but its values are not known precisely. One well-studied topic on value uncertainty is imprecise query processing. The answer to such a query is associated with a probabilistic guarantee on its correctness. There has been a growing interest in uncertain data mining.

D. *A Rule-Based Classification Algorithm for Uncertain Data[9]*

To handle uncertainty in data this method uses rule based and prediction algorithm uRule. This algorithm considers new measures computed considering uncertain data interval and probability distribution function for generating pruning and optimization. Rules extracted using uRule shows relationships between attribute and class label.

The coverage of rule gives the number of instances that satisfies the condition. The accuracy of a rule is the fraction of instances that satisfy the condition and assigned to the class label, output of a rule, normalized by condition. uLearnOneRule (), uGrow(), splitUncertain() these procedures are used to support uRule algorithm. uLearnOneRule () generates best rule for class from uncertain training set is given. This has two parts growing and pruning. splitUncertain() returns part of the instance that is covered by the rule. Initial rule of uGrow() is left side is empty and right side contains current class. Probabilistic information gain is used to select attribute and split point. When instance covered by rule, it removed from dataset as rule grows.

E. *Uncertain Neural Networks for classification [10]*

The performance and quality of data mining results are largely dependent on data uncertainty. It must be properly modeled and processed. This technique focuses on one commonly encountered type of data uncertainty. If the exact data value is unavailable and the probability distribution of the data is known then the data value is replaced by expected value. This method, although simple and straightforward, may cause valuable information loss. The conventional neural networks classifier is extended to tackle this problem so that it can take certain data and uncertain probability distribution as the input. Gaussian probability distribution is considered for this approach. UNN will perform the classification correctly since it computes the probability of P belonging to classes according to the probability distribution information and predicts it to be in the class which has a larger probability. Hence the uncertain neural network can achieve higher classification accuracy.

III. SYSTEM ARCHITECTURE

A. *Traditional Decision Tree [1][6]*

The data set consist of d training tuples ($t_{1...}$ t_d) and n numerical featured attributes (A_1... A_n). Each tuple in data set is associated with feature vector ($v_{i,1}$... $v_{i,n}$) and a class label c_i where $c_i \in C$. Feature value belongs to attribute domain. Classification problem is to construct a model that maps each feature vector to a probability distribution on class label. Test tuple $te_0 = (v_{0,1} ... v_{0,n})$ is assigned a class label with high accuracy. Each internal node is associated with an attribute A_{jn} and split point s_n which belongs to attribute domain. At each node tuples are divided into two parts "left sub tree" and "right sub tree". It is important to choose best attribute split point pair at each node to classify data effectively. Gini Index, Entropy and Information Gain are used to select best pair. Entropy and Information gain is used for this paper.

B. *System Architecture*

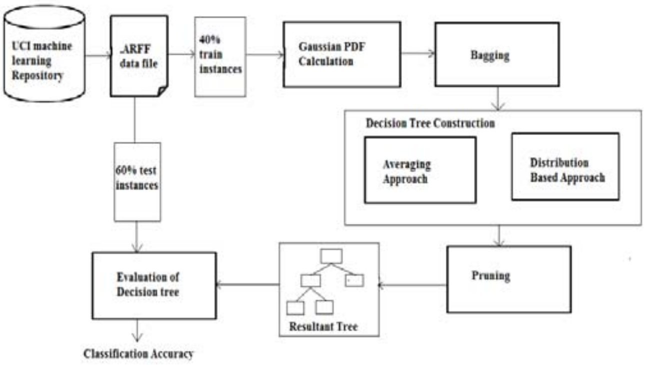

Figure 1. Figure1. System Architecure

1) *PDF Calculations[12] :*$P_r(c)$ gives the probability of how tuple is assigned a class label c at leaf node r. To determine class of given test tuple, we traverse the tree from

root to leaf. At each internal node we perform a test and compare feature value of tuple with split point then proceed towards left or right accordingly.

Feature value is not represented by single value in case of uncertainty. Probability distribution function is used to represent feature value. For practical reason we assume PDF $f_{i,j}$ is non zero value within an interval $[a_{i,j}, b_{i,j}]$. Numerical approach is assumed to calculate the PDF for rest of the paper. A set of s sample points $x\epsilon [a_{i,j}, b_{i,j}]$ is stored with associated PDF value $f_{i,j}(x)$. Gaussian PDF is used for continuous data. With this method amount of information is exploded by factor of s. Richer information allows us to build better classification model. PDF for each attribute is calculated by using:

$$P_r[a \leq x \leq b] = \int_a^b f_x(x)dx$$

$$f_x(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (1)$$

So Probability distribution for each tuple at leaf node is given as:

$$PDF(i) = \int_a^b f(x)dx = \frac{1}{\sigma\sqrt{2\pi}} \int_a^b e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (2)$$

At each internal node including root node, to determine quantity of tuples associated with a class, firstly check attribute and split point of that node n. PDF of tuple under the attribute A_{jn} lie in interval $[a_{x,jn}, b_{x,jn}]$. Left and right probability is calculated and then tuple t_x is partitioned into two sets t_L and t_R. Tuples in these subsets inherit the class labels of t_x. This concept is used in C4.5 algorithm [2].

2) Bagging[4]: Bagging method was formulated by Leo Breiman. Its name was deduced from the phrase bootstrap aggregating. Decision trees are powerful, but unstable. A small change in the training data can produce a large change in the tree. Decision tree should be stable while achieving high accuracy.

Bagging method is used to improve results of classification algorithm. Classification algorithm creates classifier based on training tuples. Bagging method first creates sequences of classifiers in respect of modification in training set. These classifiers are combined into one classifier. The prediction of such classifier is given as a weighted combination of individual classifier predictions. This approach is a base version of bagging.

$$H(d_i, c_j) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m H_m(d_i, c_j)\right) \qquad (3)$$

Some other strategies called "bagging like strategies" divide original training set into n subsets of same size. Each subset creates one classifier. A particular classifier is learned using this subset. A compound classifier is created by aggregating these particular classifiers. The most known bagging like strategies are: disjoint partitions, small bags, no replication small bags and disjoint bags. These strategies use a combination of the bagging method and the cross-validation method. In cross-validation the training set is divided into N subsets of D/N size. One of these subsets is used as the training set and rest play the role of test sets.

3) *Decision Tree Construction:* There are three main steps for decision tree construction Selection of best split point, classifying instances and pruning unwanted branches.

Selecting best split point is a crucial task in decision tree. Proper selection of split point gives more accurate results. This minimizes the degree of dispersion. Lesser dispersion means less uncertainty in data. In this paper Entropy is used as the measure of uncertainty. Minimum entropy value is expected here. Formula for calculating entropy is as follows [11]:

$$\text{Entropy}(S) = -P(\text{positive}) \log 2\, P(\text{positive}) - P(\text{negative}) \log 2P(\text{negative}) \qquad (4)$$

This gives the value of uncertainty lies in attribute value. Entropy value for attribute must be closer to zero. Minimum entropy gives the maximum Information Gain. Information gain is inverse proportion of uncertainty. Information Gain is calculated by [11]:

$$\text{InfoGain}(S, A) = \text{Entropy}(S) \sum_{v=1}^{n} \frac{|S_v|}{|S|} \times \text{Entropy}(S_v) \qquad (5)$$

Attribute with high information gain is selected for classification. Such attribute gives high accuracy of classification. We use two methods to classify instances. Averaging approach and Distribution based approach.

Averaging approach [1] transforms an uncertain data set into point valued data. It is done by replacing PDF value by its mean value. Feature vector of tuple then consist of these mean values. Decision tree is then constructed using this feature vector on traditional decision tree algorithm. As this approach uses mean values for classification pruning is not so required.

Averaging approach is a greedy approach. At each node set of training tuples S is examined. If all tuples in S has same label then make that node as leaf node and terminate the process. Otherwise take attribute and split point for that node and classify the tuples accordingly. Tuples with less value than split point goes on left side and remaining goes on right side.

This process continues for all tuples in S. In this approach single value i.e. mean of PDF is considered for classification. So there are chances of information loss. Distribution Based approach overcomes this disadvantage of averaging approach.

Distribution based [1] approach allows to use complete information carried by PDF. Instead of taking mean value all values that constitute in PDF are used to classify test tuples.

This approach utilizes more information to achieve accuracy of classifier. For training tuple t_i under attribute A_{jn} , if the pdf of tuple lie in the interval $[a_{x,jn}, b_{x,jn}]$ then tuple is divided into left and right sub tree. If $b_{x,jn} \leq$ split point then t_i is assigned to the left . If split point is less than $a_{x,jn}$ then t_i is assigned to right.

This approach is very similar to the previous one. The difference is the way tuples are classified at node. In this case split point moves in the interval so the probability changes in k steps. With m tuples there are total ks sample points and ks-1 possible split points. Considering all n attributes, to determine best split point we need to examine n(ks-1) combinations of attribute and split point. This increases number of calculations. Hence this approach is time consuming as compared to averaging approach.

Above approaches train the classifier to handle uncertain data. Averaging approach we uses mean value of PDF. So size of decision tree is automatically controlled. But for distribution based approach an interval is considered and compared tuple with each value in interval. This increased number of calculations and size of the tree. Efficiency of tree is then affected by these aspects.

Pruning techniques are used to improve efficiency of decision tree. Basic concept of pruning is removing unwanted branches from tree. Here prune the branches which span empty and homogeneous intervals.

- Definition 1 (Empty interval)[1]: An interval [a,b] is empty if

$$\int_a^b f_{i,j}(x)dx = 0 \qquad \text{for all } t_j \in S \tag{6}$$

- Definition 2 (Homogeneous interval)[1]: An interval [a,b] is homogeneous if there exists a class label c $\in$ C such that

$$\int_a^b f_{i,j}(x)dx \neq 0 \rightarrow c_j \in C \quad \text{for all } t_j \in S \tag{7}$$

Pruning will reduce the accuracy on the training data, but (in general) increase the accuracy on unseen data. It is used to mitigate over fitting, where perfect accuracy on training data would be achieved, but the model (i.e. the decision tree) is so specific that it doesn't apply to anything but that training data. In general, if pruning is increased, the accuracy on the training set will be lower.

Global Pruning algorithm is very effective in pruning intervals. GP reduces the number of entropy calculations including the calculation of entropy values of the split points. Only the candidate split points that give suboptimal entropy values are pruned away. So, even after pruning, there need of 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.

IV. EXPERIMENTAL RESULTS

The first divide dataset taken from UCI machine Learning Repository into train dataset and test dataset. We considered various combinations of train and test datasets. Train dataset is used for decision tree construction. Test dataset is used for evaluation. We used Gaussian Probability Distribution Function to handle uncertain data. We considered 100 sample points per PDF as baseline settings.

We implemented averaging and distribution based approach to explore the potential of decision tree classifier for uncertain data. We applied these approaches on real data sets taken from UCI Machine Learning Repository. These data sets are chosen because they contain uncertain data from measurement errors.

We applied ID3; C4.5 and Random Tree algorithms on these datasets .These algorithms are used for both the approaches i.e. Averaging and Distribution based approach. It is observed that among ID3, C4.5 and Random Tree algorithms C4.5 is better. We used C4.5 for all approaches.

TABLE 1. Accuracy of different uncertain datasets

Datasets	***Averaging Approach***		***Distribution Based Approach***	
	C4.5	***C4.5 with Bagging***	***C4.5***	***C4.5 with Bagging***
Japanese Vowel	***83.73%***	***83.06%***	***83.06%***	***83.06%***
Page Block	***97.82%***	***97.00%***	***97.20%***	***97.80%***
Segment	***98.11%***	***94.78%***	***95.60%***	***95.60%***
Vehicle	***86.02%***	***82.87%***	***73.42%***	***73.62%***
Ionosphere	***92.89%***	***92.00%***	***90.04%***	***93.83%***
Iris	***100%***	***100%***	***98.88%***	***100%***

1) Averaging Approach:

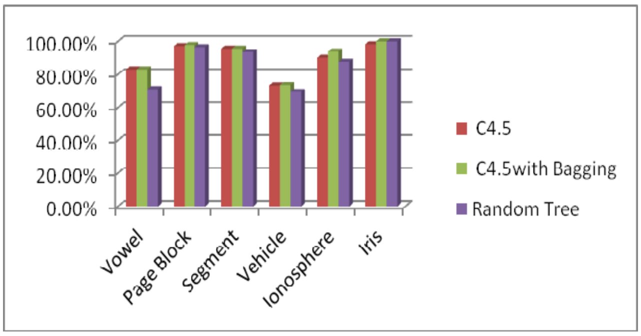

Figure 2. Accuracy of uncertain datasets (Averaging Approach)

2) Distribution Based Approach:

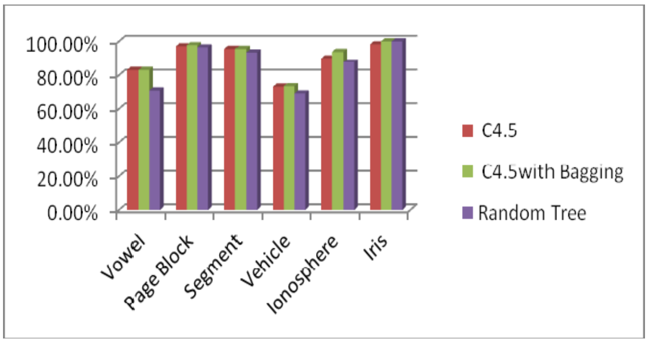

Figure 3. Accuracy of uncertain datasets (Distribution Based Approach)

If Data sets are divided into 40%train data and 60% test data, we get moderate accuracy. Train data set is assumed to be classified so percent of train dataset is kept less. These datasets are applied on various algorithms for averaging and distribution based approaches. It is observed that averaging approach is less time consuming than Distribution based approach as it has less entropy calculations. So the accuracy of this approach is less than distribution based approach. Stability of decision tree classifier is achieved by using Bagging method. It also improves accuracy of classifier. The graph shows the comparison on basis of accuracy.

V. CONCLUSION

Potential of Decision tree classifier can be used to accommodate data tuples with uncertain data. Enhancing decision tree with data uncertainty measures such as information entropy and information gain, Gaussian PDF gives better accuracy. When suitable PDFs are used decision tree gives remarkably higher accuracy. Performance of decision tree while handling uncertain data is an issue, because of the increased amount of information to be processed.

By combining bagging technique and pruning techniques stable performance can be achieved. Pruning techniques reduces excess execution time required to process empty and homogeneous intervals. It is cleared from experimental results and graphs that 1) distribution approach is better than averaging approach as it uses full information associated with PDF. 2) Bagging method improves results as well as it improves performance of decision tree.

REFERENCES

[1] Smith Tsang, Ben Kao, Kevin Y. Yip, Wai-Shing Ho, and Sau Dan Lee, “A.Decision Trees for Uncertain Data,” AI KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 1, pp. 64- 78 , JANUARY 2011.

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

[3] N.N. Dalvi and D. Suciu, E_cient Query Evaluation on Probabilistic Databases, The VLDB J., vol. 16, no. 4, pp. 523-544,2007.

[4] Kristína Machova, Frantisek Barcak, Peter Bednar “A Bagging Method using Decision Trees in the Role of Base Classifiers”, Department of Cybernetics and Artificial Intelligence, Technical University, Letna, Acta Polytechnica Hungarica Vol. 3, No. 2, 2006.

[5] G.V.SURESH, E.V.Reddy, Shabbeer Shaik “*Classification of Uncertain Data using Gaussian Process Moel,*” Indian Journal of Computer Science and Engineering Vol. 1 No. 4 306-312.

[6] Biao Qin , Yuni Xia , Fang Li, “*DTU: A Decision Tree for Uncertain Data,*” Department of Computer and Information Science, Indiana University – Purdue.

[7] Michalis Vazirgiannis, Maria Halkidi*,” Uncertainty handling in the data mining process with fuzzy logic,*” Department of Informatics Athens University of Economics & Business Patision 76, 10434, Athens, Greece (Hellas). Data Mining (ICDM), pp. 436-445, Dec. 2006.

[8] W.K. Ngai, B. Kao, C.K. Chui, R. Cheng, M. Chau, and K.Y.Yip, “Efficient Clustering of Uncertain Data,” Proc. Intl Conf. Data Mining (ICDM), pp. 436-445, Dec. 2006.

[9] Thair Nu Phyu,” Survey of Classification Techniques in Data Mining,” Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol IIMECS 2009, March 18 - 20, 2009, Hong Kong.

[10] Jiaqi Ge, and Yuni Xia, UNN: A Neural Network for uncertain data classification, Department of Computer and Information Science, Indiana University – Purdue University, Indianapolis, USA

[11] J.R. Quinlan, “Induction of Decision Trees,” AI Machine Learning, vol. 1, no. 1, pp. 81-106, 1986.

[12] Kishor Trivedi, "Probability Models for Computer Science," First Edition,Academic Press, San Diego,CA.

[13] http://www.archive.ics.uci.edu/ml/