Classification: A Decision Tree For Uncertain Data Using CDF

VarshaChoudhary[1], Pranita Jain[2]
(Department Of Information Technology,SATI College, RGPV University
Vidisha (M.P) , India ,)
(Ass. Prof.InDepartment Of Information Technology,SATI College, RGPV University
Vidisha (M.P) , India,)

ABSTRACT

The Decision trees are suitable and widely used for describing classification phenomena. This paper present a decision tree based classification system for uncertain data. The uncertain data means lack of certainty. Data uncertainty comes by different parameters including sensor error, network latency measurements precision limitation and multiple repeated measurements. We find that decision tree classifier gives more accurate result if we take "complete information" of data set .In this paper we improve the traditional decision tree algorithm which works on with known and precise data , including gini index method for determining the goodness of a split and considering cumulative distribution function . The experimental study shows that proposed CDF-distribution based algorithm gives accurate result for uncertain numerical dataset and it is computationally efficient in terms of memory, time and accuracy.

Keywords–Classification,CDF,Datamining ,Decision tree ,Uncertain data .

I. INTRODUCTION

The Data mining refers to extracting or mining knowledge from large amounts of data. The classification of large data set is an important problem in data mining . The classification problem can be defined as follows for a database with a number of records and for a set of classes such that each record belongs to one of the given classes , the problem of classification is to decide the class to which given record belongs. The classification problem is also concerned with generating a description a model for each class from the given data set. Classification is one of the most important data mining techniques .It is used to predict group/class membership for data instances.

Different models have been proposed for classification such as Decision tree, Neural networks ,Bayesian belief networks, Fuzzy set and Genetic models. The decision trees classifier are most widely used in among of these models for classification. They are popular because they are practically and easy to understand. Rules can also be extracted from decision trees easily. Many algorithm such as ID3[7] ,C4.5 and CART have been devised for decision tree construction . All of these algorithms are used in various areas such as image recognition , medical diagnosis[5] ,credit rating of loan applicants , scientific tests , fraud detection and target marketing. The decision tree is a supervised classification approach .A decision tree is a flow chart like structure , where each internal node denotes a test on an attribute , each branch shows an outcome of the test and each leaf node holds a class label. The top node in a tree is define a root node . A decision tree has a two different sub sets – a training set and a test set. The training set is used for deriving the classifier and test set is used to measure the accuracy of the classifier. The accuracy of the classifier is determined by the percentage of the test data set that is correctly classified.

A decision tree works on two different kind of attributes namely numerical and categorical . Those attribute which works on numeric data known as numerical attribute and the attributes whose domain is not numeric are called the categorical attributes. The aim of classification is to design a concise model that can be used to predict the class of the data records whose class label is unknown.

A simple way to handle uncertainty is to abstract probability distribution by summary statistics such as means and variance. This approach is known as Averaging. Another method is works on the complete information carried by the probability distributions to design a decision tree. This method is known as distribution based[1]. In this paper we works on distribution based method with "cumulative distribution function " (cdf) for constructing decision tree classifier on uncertain numerical data sets.

A uncertainty comes on many application due to different reasons. We shortly describe different kind of uncertainty here:-

1.1.Parameter uncertainty **:-** A parameter uncertainty which comes from the model parameter that are inputs to the computer model (mathematical model)but whose exact values are unknown to experimentalists and cannot be controlled in physical experiments.

1.2.Structural uncertainty **:-** This type of uncertainty comes from the lack of knowledge of the underlying true physics . It depends on how accurately a mathematical model describes the true system for a real life situation , considering the fact that models are almost always only approximations to really.

1.3.Experimental uncertainty **:-** This type of uncertainty comes from the variability of experimental measurements. The experimental uncertainty is inevitable and can be noticed by repeating a measurement for many times using exactly the same setting for all inputs/variables.

In this paper our contributions include :

1. A basic algorithm for building decision trees for uncertain numerical datasets.

2.A experimental study compare the classification accuracy achieved by the UDT based on pdf and UDT based on cdf .

3.A performance analysis based on cpu time , memory for both algorithm.

In the rest of this paper section 2 describes related work , section 3 describes problem definition , section 4 shows the proposed algorithm UDT-CDF and section 5 shows experimental study on both algorithm. The last section we concludes the paper.

II. Related Work

There are many uncertain data classification algorithms have been proposed in the literature in recent years. Qin et al(2009b) proposed a rule –based classification algorithm for uncertain data[4]. Ren et al.(2009) proposed to apply Naïve Bayes approach to uncertain data classification problem. A decision tree is widely used classification models because of its advantage(Tsang et al 2009 [1],Quinlan 1993[2]). Various decision tree based classifiers for uncertain data are proposed by research. The C4.5 classification algorithm were extended to the DTU(Qin et al 2009a)[2] and the UDT (Tsang et al 2009)[1] for classifying uncertain data.(Qin et al 2009a) used probability vector and probability density function (pdf) to represent uncertain numerical attribute (Qin et al 2009b) and uncertain numerical attribute (Cheng et al 2003) respectively. They constructed a well performance decision tree for uncertain data (DTU).A C.Liang , Y.Zhang(2010) proposed an algorithm UCVFDT which works on dynamic and uncertain data streams [3] .Tsang et al (2009)[1] used the "complete information " for pdf to construct a uncertain decision tree (UDT) and proposed a series of pruning techniques to improve the efficiency.

In our algorithm we use a cumulative distribution function to construct a uncertain numerical decision tree and it gives more accurate result compare to UDT which works on pdf[1]. A cumulative distribution function is basically probability based distribution function .

Another related topic is Fuzzy decision tree[13] . Fuzzy information models data uncertainty arising from human perception and understanding. The uncertainty reflects the vagueness and ambiguity of concepts, e.g., how cool is "cool". In fuzzy decision tree, both attributes and class labels can be fuzzy and are represented in fuzzy terms[1]. Given a fuzzy attribute of a data tuple, a degree (called membership) is assigned to each possible value, showing the extent to which the data tuple belongs to a particular value. Our work instead gives classification results as a distribution: for each test tuple, we give a distribution telling how likely it belongs to each class .There are many variations of fuzzy decision trees, e.g., fuzzy extension of ID3[13] and Soft Decision Tree[14]. In these models, a node of the decision tree does not give a crisp test which decides deterministically which branch down the tree a training or testing tuple is sent. Rather it gives a "soft test" or a fuzzy test on the point-valued tuple. Based on the fuzzy truth value of the test, the tuple is split into weighted tuples (akin to fractional tuples) and these are sent down the tree in parallel[1]. This differs from the approach taken in this paper, in which the probabilistic part stems from the uncertainty embedded in the data tuples, while the test represented by each node of our decision tree remains crisp and deterministic. The advantage of our approach is that the tuple splitting is based on probability values, giving a natural interpretation to the splitting as well as the result of classification.

III. Problem Definition

This section focus on the problem of decision-tree classification on uncertain data. We describe traditional decision trees in shortly. Then, we discuss how data tuples with uncertainty are handled.

3.1.Traditional Decision Trees :

In our model, a dataset consists of d training tuples, $\{t_1,t_2, \ldots, t_d\}$ and k numerical (real-valued) feature attributes, $A_1,\ldots,A_k$. The domain of attribute Aj is dom(Aj). Each tuple t_i is associated with a feature vector Vi =$(v_{i,1}, v_{i,2},.. v_{i,k})$ and a class label c_i, where $v_{i,j}$ €dom(Aj) and c_i € C, the set of all class labels. The classification problem is to construct a model M that maps each feature vector $(v_x,1,\ldots,.. v_x,k)$ to a probability distribution Px on C such that given a test tuple $t_0 = (v_{0,1},\ldots, v_{0,k}, c_0)$, $P_0 = M(v_{0,1},\ldots, v_{0,k})$ predicts the class label c0 with high accuracy. We say that P_0 predicts c_0 if c_0 = arg max c €C P_0 (c).[1]

In this paper we study binary decision trees with tests on numerical attributes. Each internal node n of a decision tree is associated with an attribute and a split point .An internal node has exactly 2 nodes, which are labeled "left" and "right", respectively .Each leaf node in a binary tree associate with class label.

To determine the class label of a given test tuple t_0 ,we traverse the tree starting from the root node until a leaf node is reached. When we visit an internal node n, we execute the test and proceed to the left node or the right node accordingly. Eventually, we reach a leaf node m. The probability distribution Pm associated with m gives the probabilities that t_0 belongs to each class label c € C. For a single result, we return the class label c € C that maximizes Pm(c).

3.2.Handling uncertainty:

In our algorithm , a feature value is represented not by a single value , but by a cdf . In our algorithm the cumulative distribution function (CDF) describes probability of a random variable falling in the interval (−∞, *x*].The CDF of the standard normal distribution is denoted with the capital Greek letter Φ (phi), and can be computed as an integral of the probability density function:

$$p = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}}\,dt \qquad \textbf{eq.(1)}$$

Numerical methods for calculation of the standard normal CDF are as follows For a generic normal random variable with mean μ and variance $\sigma^2 > 0$ the CDF will be equal to

$$F(x;\mu,\sigma^2) = \Phi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{2}\left[1+\operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right], \quad x \in \mathbb{R}.$$

eq.(2)

3.3.Propertiesof cdf :

3.3.1 The standard normal CDF is 2-fold rotationally symmetric around point (0, ½): $\Phi(-x) = 1 - \Phi(x)$.

3.3.2 The derivative of $\Phi(x)$ is equal to the standard normal pdf $\phi(x)$: $\Phi'(x) = \phi(x)$.

3.3.3 The antiderivative of $\Phi(x)$ is: $\int \Phi(x)\,dx = x\,\Phi(x) + \phi(x)$

IV. Algorithm for UDT-CDF

Input : the training dataset DS (Japanese vowel) ; the set of candidate attributes att-list

Output : An uncertain numerical tree

```
Begin
4.1 create a node N;
4.2  if (DS are all of the same class, C)then
4.3  return N as a leaf node labeled with the class C;
4.4 else if (attribute –list is empty)then
4.5  return N as a leaf node labeled with the highest weight class in DS;
4.6 endif;
4.7 select a test-attribute with the highest probabilistic information gini index to label node N;
4.8 if (test-attribute is uncertain numeric ) then
4.9 binary split the data from the selected position p;
4.10for (each instance i)do
4.11 if(test-attribute <= p)then
4.12 put it into DS_l left side with weight i.w;
4.13 Else if (test-attribute>p)then
4.14 Put it into DS_r right side with weight i.w;
4.15 else
4.16 put it into DS_L with weight i.w(1)*
```

$$p = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}}\,dt \qquad \textbf{eq.(1)}$$

```
4.17 put it into DS_R with weight i.w(2)*
```

$$p = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}}\,dt \qquad \textbf{eq.(1)}$$

```
4.18 endif;
4.19 end for;
```

The basic concept of this algorithm as follows:

(1)The tree starts as a single node representing the training samples (step 1).

(2) If the samples are all of the same class; then the node becomes a leaf and is labeled with that class (step2 and 3).

(3) Otherwise, the algorithm uses a probabilistic measure, known as the probabilistic information gini index, as the criteria for selecting the attribute that will best separate the samples into an individual class (step 7). This attribute becomes the ”test” attribute at the node.

(4) If the test attribute is uncertain numerical, we split for the data at the selected position p (steps 8 and 9).

(5) A branch is created for test-attribute ≤ p or test-attribute > p respectively. If an instance’s test attribute value is less than or equal to p , it is put into the left branch with the instance’s weight i .w(1). If an instance’s test attribute value is larger than p , it is put into the right branch with the instance’s weight i .w(2). If an attribute’s value covers the split point p (-∞ ≤ p <∞), it is put into the left branch with weight i .w∗ cdf eq.

$$p = F(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}}\,dt \qquad \textbf{eq.(1)}$$

(6) And for right branch with weight i .w∗cdfeq(1). Then the dataset is divided into Ds_l and DS_r (steps 10-19).

(7) The algorithm recursively applies the same process to generate a decision tree for the samples.

(8) The recursive partitioning process stops only when either of the following conditions becomes true:

(8.1) All samples for a given node belong to the same class (steps 2 and 3),

Or

(8.2) There are no remaining attributes on which the samples may be further partitioned (step 4). In this case, the highest weight class is employed (step 5). This involves converting the given node into a leaf and labeling it with the class having the highest weight among samples. Alternatively, the class distribution of the node samples may be stored.

V. Experiments

In this section , we present the experimental results of the proposed decision tree algorithm .we have implemented UDT based on pdf[1]and UDT based on cdf and applied them to real data sets which name is Japanese vowel .it is taken from the UCI Machine Learning Repository[11]. This data set chosen because this contain mostly numerical attributes obtained from measurements.
The Japanese vowel data set have 640tuples, in which 270 are training tuples and 370 are the test tuples .Each tuple representing anutterance of Japanese vowels by one of the 9 participatingmale speakers. Each tuple contains 12 numerical attributes, which are LPC (Linear Predictive Coding) coefficients. These coefficients reflect important features of speech sound. Each attribute value consists of 7–29 samples of LPC coefficients collected over time. These samples represent uncertain information and are used to model the cdf of the attribute for the tuple. The class label of each tuple is the speaker id. The classification task is to identify the speaker when given a test tuple.

We implemented UDT-CDF on MATLAB 7.8.0(R2009a) , the experiments are executed on pc with Intel (R)Pentium(R) ,2.30GHZ CPU and 2.00 GB main memory .

5.1 Accuracy:

The overall accuracy of calculate in this graph.
(The accuracy of the classifier is determined by the percentage of the test data set that is correctly classified).We first examine the accuracy of the algorithms , which is shown in figure 1.

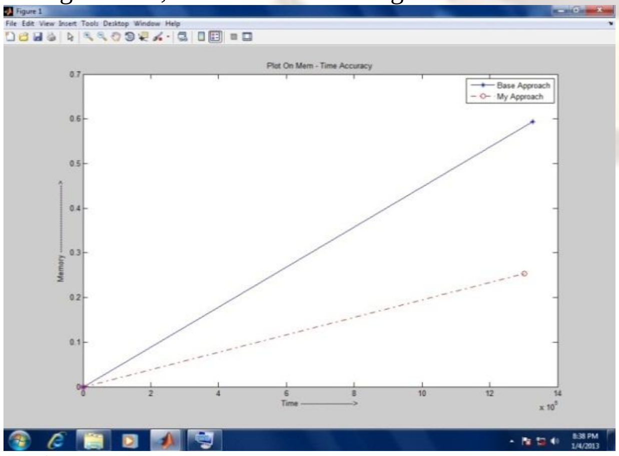

fig: 1 UDT-CDF accuracy on uncertain numerical data sets

In figure x-axis defines time and y-axis defines memory .the dotted line indicates accuracy achieved by UDT-CDF .

5.2 Execution time:

The Execution time or CPU time of a algorithm is defined as the time spent by the system executing that particular algorithm, including the time spent executing run-time or system services on its behalf. we examine the execution time of the algorithms, which is shown in figure 2.

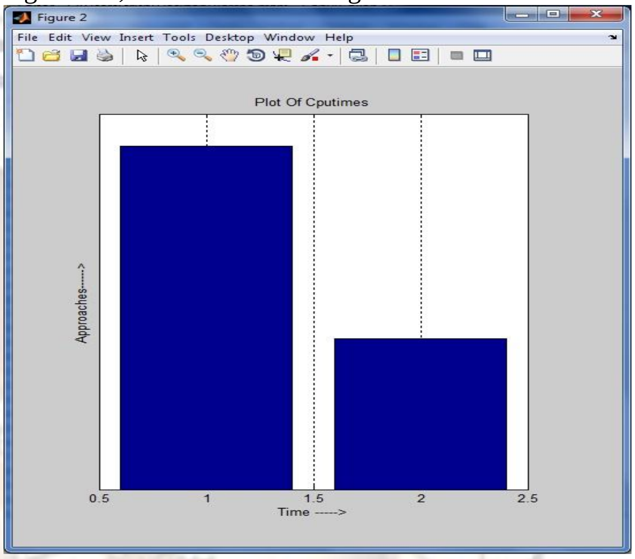

fig: 2 UDT-CDF cputime on uncertain numerical data sets

The diagram show cputime (in seconds) or execution time of UDT-PDF and UDT-CDF .where x-axis defines time and y-axis defines pdf, cdf algorithms. Our proposed algorithm takes less time for execution of uncertain numerical datasets. The first bar defines execution time of udt-pdf and second bar defines the execution time taken by udt-cdf.

5.3Memory

We examine the memory space used by the algorithms , which is shown in figure 3.

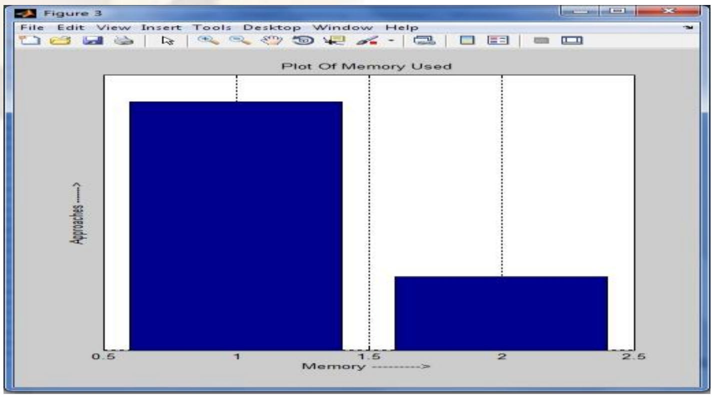

fig: 3 Memory graph

while we use complete information for plot a uncertain decision tree then we need more memory space for storing the information. The figure 3 memory graph defines memory space used by UDT-PDF and UDT-CDF. Where first bar for UDT-PDF and second graph for UDT-CDF.

5.4 Classification diagram :

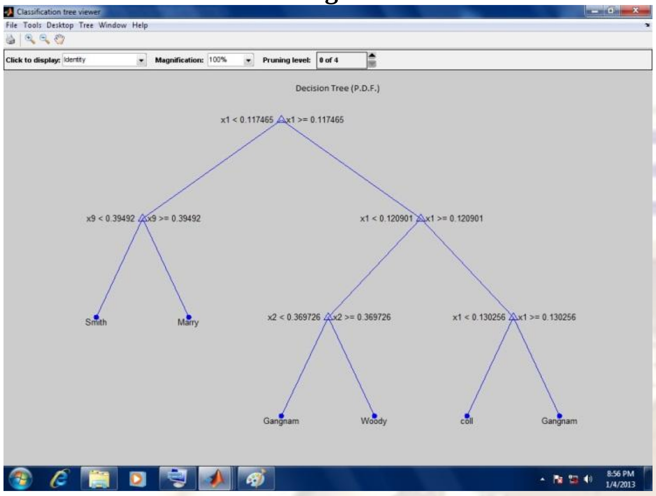

fig:4 UDT-PDF decision tree

Rules for classification of UDT-PDF:
1 if x1<0.117465 then node 2 else node 3
2 if x9<0.39492 then node 4 else node 5
3 if x1<0.120901 then node 6 else node 7
4 class = Smith
5 class = Marry
6 if x2<0.369726 then node 8 else node 9
7 if x1<0.130256 then node 10 else node 11
8 class = Gangnam
9 class = Woody
10 class = coll
11 class = Gangnam

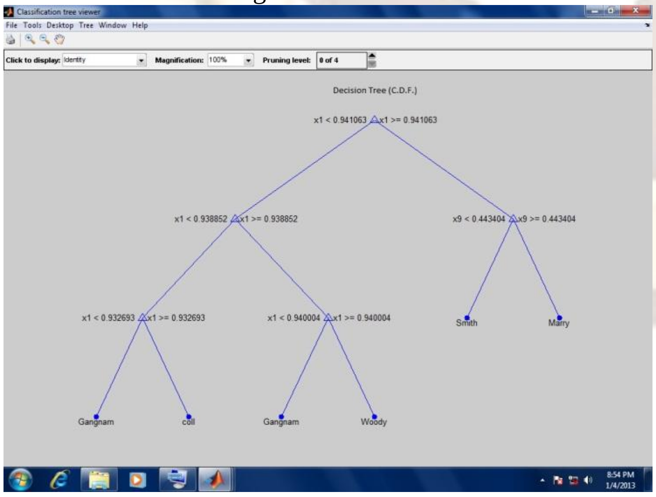

fig:5UDT-CDF decision tree

Rules for classification UDT-CDF:
1 if x1<0.941063 then node 2 else node 3
2 if x1<0.938852 then node 4 else node 5
3 if x9<0.443404 then node 6 else node 7
4 if x1<0.932693 then node 8 else node 9
5 if x1<0.940004 then node 10 else node 11
6 class = Smith
7 class = Marry
8 class = Gangnam
9 class = coll
10 class = Gangnam
11 class = Woody

Fig4.and 5 shows the diagram of decision tree of UDT-PDF and UDT-CDF .The classification rules describe how the classified .

The table1.1 shows the result analysis of both algorithm.

Dataset Use	Total no.of Blocks	Total no.of Users	Approach Used	Execution accuracy		
				Memory Used	Total time	
					Overall	Cpu Access
ae.train.dat	270	9	P.D.F.	1.876 K.B.	1.360 s	2.68 m.s.
			C.D.F.	1.275 K.B.	1.001 s	2.13 m.s.
ae.test.dat	370	9	P.D.F.	2.120 K.B.	1.520 s	3.68 m.s.
			C.D.F.	1.925 K.B.	1.223 s	3.452 m.s.

From the table-1.1 we see that UDT-CDF builds more accurate decision trees than UDT[1]. The Japanese vowel dataset is used for experiment which contain two sets firstly training dataset(ae.train.dat) and second is testing dataset (ae.test.dat). A training set has 270 blocks and test set has 370 blocks. The total numbers of user are 9.The total memory used by UDT-CDF is 1.275K.B while UDT-PDF takes 1.876K.B for same training set. We see that not only UDT-CDF takes less memory for training set but also for test set. The total execution time of both algorithm is define in table , where UDT-CDF takes less cputime for execution.

VI. Conclusion

We have implemented a new algorithm UDT-CDF for classification of uncertain numerical data sets. Our algorithm gives a novel result in terms of memory, execution time and accuracy. Our algorithm can able to work on discrete random variables data sets and continuous random variables data sets.

References

[1] S.Tsang ,B.Kao, K.Y.Yip, W-S Ho and S-D.Lee; (2009) *Decision Trees For Uncertain Data* in IEEE.

[2] B.Qin ,Y.Xia and F.Li ;(2009) , *DTU:A decision tree for classifying uncertain data* the PAKDD pp,4-15.

[3] C.Liang, Y.Zhang and Q.Song ;(2010), *Decision tree for Dynamic and Unceratin Data Strems* JMLR: workshop and conference proceeding 13:209-224.

[4] B.Qin ,Y.Xia ,S.Prabhakar and Y.Tu ;(2009) *A rule-based classification algorithm for uncertain data:* the proc the IEEE workshop on Management and Mining of Uncertain Data (MOUND) .

[5] C.L.Tsien , I.s. Kohane and N.Mclntosh ,*Multiple signal integration by decision tree induction to detect artifacts in the neonatal intensive care unit.* Artificial intelligence In Medicine , vol.19 , no.3, pp.189-202, 2000.

[6] J.Gama, P.Medas and P.Rodrigues .*Learning Decision Trees from Dynamic data streams* . Journal of Universal Computer Science, 11(8):1353-1366,2005.

[7] J. Gama, R.Fernandes and R.Rocha.*Decision trees for mining data streams* .Intell . Data Anal ,1:23c-45,2006.

[8] J.R. Quinlan .*Induction of decision trees*. Machine Learning, Vol-1, no.1, pp.81-106.0986.

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

[10] 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.

[11] A.Asuncion and D.Newman. *UCI machine learning repository* .2007. [online].

[12] C.K.Chui, B.Kao and E.Hung. *Mining frequent itemsetsfromm uncertain data*. In PAKDD ser. Lecture notes in computer science. Vol.4426. Nanjing ,China : Springer ,22-25 May 2007 ,pp .47-58.

[13] C.Z Janikow, *Fuzzy decision tree: Issues and mehods* .IEEE Transactions on Systems, Man , and cybernetics , Part B, vol.28. no.1, pp.1-14, 1998.

[14] C.Olaru and L. Wehenkel ,*A complete fuzzy decision tree technique* . Fuzzy sets and Systems.Vol 138, no.2, pp. 221-254,2003.