MODIFIED GINI INDEX CLASSIFICATION: A CASE STUDY OF HEART DISEASE DATASET

N.SUNEETHA[#1] CH.V.M.K.HARI[$2] V.SUNIL KUMAR
Department of IT, GITAM University [#1, $2 *3]
suneetha_sun50@yahoo.co.in ,kurmahari@gitam.edu , sunil.veee@gmail.com ,

Abstract: Classification has been used for predicting medical diagnosis. Classification methods can handle both numerical and categorical attributes. Gini index uses the method which biases multivalued attributes. When the number of classes are large, and the biases are increased, the Gini-based decision tree method is modified to overcome the known problems, by normalizing the Gini indexes by taking into account information about the splitting status of all attributes. Instead of using the Gini index for attribute selection ratios of Gini indexes are used and their splitting values in order to reduce the biases. Experiments are done on heart diseases dataset and Report of experimental graph is shown by comparing between the modified method and other known classification algorithms ID3, c4.5, Generalized Gini Index classifies relevant parts into various groups

Keywords: GINI INDEX, Classification, Medical Diagnosis, Data Mining, ID3((Iterative Dichotomiser 3).

INTRODUCTION:

A **decision tree** (or tree diagram) is a decision support tool that uses a graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. A decision tree is used to identify the strategy most likely to reach a goal. Another use of trees is as a descriptive means for calculating conditional probabilities.

Decision tree technique is most widely used among all other classification methods. The popularity is due to conceptual transparency, inexpensive computation, ease of interpretation and robustness etc., Decision tree analysis is a technique that classifies the relevant parts into various groups. The decision tree methodology was developed by Breiman, Friedman, Olshen and Stone.This method introduces a classification /regression tree for a univariate discrete/continuous response. There are various competing approaches to the work of Breiman et al from Hawkins and Kass, and Quinlan . All of these approaches focus on a single response. In the past decade, researchers have constructed decision trees for multiple responses. Segal and Zhang suggested a tree that can analyze continuous longitudinal responses.

Decision tree classification has been used for predicting medical diagnoses. Among data mining methods for classification, decision trees have several advantages such as they are simple to understand and interpret; they are able to handle both numerical and categorical attributes. Even though most algorithms for decision tree give accurate models for prediction, none is significantly superior than others. There is usually no clear-cut best algorithm. Most existing algorithms find models that fit some static model. Although accurate in some cases, these algorithms can break down in other cases. That is, they may predict the data incorrectly. It is well-known that when Gini index is used for classification, the method biases multivalued attributes. In addition to having difficulty when the number of classes is large, the method also tends to favour tests that result in equal-sized partitions and purity in all partitions.

Modifying the Gini-based decision tree method. Instead of using the Gini index for attribute selection as usual, we use ratios of Gini indexes in order to reduce the biases. We report our experiments with several benchmark medical data-bases. Comparisons between our method and other known classification algorithms are provided. Since Gini ratios can be calculated during the calculation of Gini indexes, the time complexity and space complexity of the modified algorithm remain the same as the complexities of Gini index algorithm. analyses and experiments with heart diseases medical data base is presented. It is well –known that when gini-index is used for classification, the method biases used multivalued attributes.

METHODOLOGY

Different Classification Models

ID3((Iterative Dichotomiser 3)

This algorithm is used to generate a decision tree invented by Ross Quinlan for introducing one of the classification models, from data.

The basic ideas behind ID3 are that:

In the decision tree each node corresponds to a non-categorical attribute and each arc to a possible

value of that attribute. A leaf of the tree specifies the expected value of the categorical attribute for the records described by the path from the root to that leaf. In the decision tree at each node should be associated the non-categorical attribute which is most informative among the attributes not yet considered in the path from the root. Entropy is used to measure how informative is a node. The algorithm is formalized using the concept of information entropy:

$$I_E(i) = -\sum_{j=1}^{m} f(i,j) \log_2 f(i,j).$$

The ID3 algorithm can be summarized as follows:

1. Take all unused attributes and count their entropy concerning test samples
2. Choose attribute for which entropy is maximum
3. Make node containing that attribute

ID3 (Examples, Target_Attribute, Attributes)

Create a root node for the tree
If all examples are positive, Return the single-node tree Root, with label = +.
If all examples are negative, Return the single-node tree Root, with label = -.
If number of predicting attributes is empty, then Return the single node tree Root, with label = most common value of the target attribute in the examples.
Otherwise Begin
A = The Attribute that best classifies examples.
Decision Tree attribute for Root = A.
For each possible value, vi, of A,
Add a new tree branch below Root, corresponding to the test A = vi.
Let Examples(vi), be the subset of examples that have the value vi for A
If Examples(vi) is empty
Then below this new branch add a leaf node with label = most common target value in the examples
Else below this new branch add the subtree ID3 (Examples(vi), Target_Attribute, Attributes – {A})
End
Return Root

3.2. Decision Tree using C4.5 Algorithm:

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set S= sbs2,... of already classified samples. Each sample Si= X1,X2,... is a vector where xI,x2,... represent attributes or features of the sample. The training data is augmented with a vector C= cbc2,... where cbc2,... represent the class that each sample belongs to.

C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller subsists.

This algorithm has a few base cases; the most common base case is when all the samples in your list belong to the same class. Once this happens, you simply create a leaf node for your decision tree telling you to choose that class. It might also happen that none of the features give you any information gain, in this case C4.5 creates a decision node higher up the tree using the expected value of the class. It also might happen that you've never seen any instances of a class; again, C4.5 creates a decision node higher up the tree using expected value.

Improvements from ID3 algorithm

C4.5 made a number of improvements to ID3. Handling both continuous and discrete attributes - In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those whose attribute value is above the threshold and those that are less than or equal to it. Handling training data with missing attribute values - C4.5 allows attribute values to be marked as ? for missing. Missing attribute values are simply not used in gain and entropy calculations. Handling attributes with differing costs.Pruning trees after creation - C4.5 goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes. C4.5 is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on.

C4.5 Algorithm

Check for base cases
For each attribute ’a’
Find the normalized information gain from splitting on ‘a’
Let ‘a’ best be the attribute with the highest normalized information gain
Create a decision node that splits on a best
Recur on the sublists obtained by splitting on a best and add chose nodes as children of node.

Information Gain Calculation

Select the attribute with the highest information gain Let pi be the probability that an arbitrary tuple in D belongs to class Ci, estimated by |Ci, D|/|D| Expected

information (entropy) needed to classify a tuple in D:
Information needed (after using A to split D into v partitions) to classify D: Information gained by branching on attribute A

$$Gain\ (A) = Info(D) - Info_A(D)$$

$$Info\ (D) = -\sum_{i=1}^{m} p_i \log_2 (p_i)$$

Decision Tree Using Gini index:

Gini index builds decision trees from a set of training data in the same way as id3, using the concept of information entropy. The training data is a set s = s1,s2,... of already classified samples. Each sample si = x1,x2,... is a vector where x1,x2,... represent attributes or features of the sample. The training data is augmented with a vector c = c1,c2,... where c1,c2,... represent the class that each sample belongs to.Gini index uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets.

Gini index examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sub lists. This algorithm has a few base cases, the most common base case is when all the samples in your list belong to the same class. Once this happens, you simply create a leaf node for your decision tree telling you to choose that class.

It might also happen that none of the features give you any information gain, in this case gini index creates a decision node higher up the tree using the expected value of the class. it also might happen that you've never seen any instances of a class;again, gini index creates a decision node higher up the tree using expected value.

Attribute Selection

Consider a N labeled class pattern partitioned into sets of patterns belonging to classes Ci ,i=1,2,3,..l. The population in class Ci is ni .Each pattern has n features and each feature can take two or more values.

For each attribute the Entropy is calculated using the formula:

$$Entropy(C) \equiv H(C) \equiv -\sum_{c} P(C=c)\log_2 P(C=c)$$

Measuring Entropy:

Entropy of class C of set of examples S is

$$H(C) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

where $p_+ = \frac{p_i}{p_i+n_i}$; p_i positive and n_i negative examples in S

Entropy can be used to measure gain in information about class by branching on attribute a – compute reduction in entropy of class c caused by knowing a value

Information Gain

$$Gain(C,A) = I(C,A) = H(C) - \sum_{v \in V(A)} P(A=v)H(C|A=v)$$

This measures the information between A and C : the amount of information we learn about C by knowing the value of A

Normalizing Entropy Gain

D

This measures entropy of S wrt V(A) (instead of wrt to class C)

$$GainRatio(S,A) = Gain(S,A)/SplitInformation(S,A)$$

Disadvantage Of Gini-index

In traditional Gini index classification, the Gini index measure is used as a heuristic for selecting the attribute that will best partition the training tuples into individual classes. The selected attribute is then used as the testing one at the node of the tree.Let D be a database consisting of |D| = d data tuples. Assume that the class label
attribute has n distinct values representing n different classes C1,C2 . . .Cn. gini(D) is defined as

$$gini(D) = 1 - \sum_{i=1}^{n} p_i^2$$

where $p_i = \frac{|C_i|}{d}$ is the relative frequency of class C_i in D.

For an attribute A with m distinct values, the database D is partitioned into m subsets
D1,D2 . . .Dm. The Gini index of D with respect to the attribute A is defined as

$$gini_A(D) = \sum_{i=1}^{m} \frac{|D_i|}{d} \cdot gini(D_i)$$

The reduction in impurity of D with respect to the attribute A is defined as

$$\Delta gini(A) = gini(D) - gini_A(D).$$

In traditional Gini-based classification, the attribute provides the the largest reduction in impurity is chosen to split the node. However, it is well known that the method biases multivalued attributes. In addition to having difficulty when the number of classes is large, the method also tends to favor tests that result in equal-sized partitions and purity in all partitions.

To overcome these known problems, we normalizethe Gini indexes by taking into account information about the splitting status of subsets D1,D2 . . .Dm. The splitting status of D with respect to the attribute A is calculated as

$$split_A(D) = 1 - \sum_{i=1}^{m} (\frac{|D_i|}{d})^2.$$

Modified Gini Index

While modifing the Gini-based decision tree method. Instead of using the Gini index for attribute selection as usual, we use ratios of Gini indexes in order to reduce the biases Gini ratios can be calculated during the calculation of Gini indexes, the time complexity and space complexity of the modified algorithm we normalize the Gini indexes by taking into account information about the splitting status of subsets The Splitting Equation for Modified Gini Index

$$split_A(D) = 1 - \sum_{i=1}^{m} (\frac{|D_i|}{d})^2.$$

The Gini ratio of D with respect to the attribute A is defined as

$$giniRatio(A) = \Delta gini(A)/split_A(D)$$

Modified Gini Index Decision Tree Algorithm

Input: The training database D
Output:A decision tree
Step1: Create a node N

• If D are all of the same class C then return N as a leaf node with the class C.

• If D has no Non-label attribute then return N as a leaf node with the most common class.

Step2: Select an attribute, say A, with the highest Gini ratio value. Label node N with A.

Step3: Partition the database D into subsets D1,D2 ..Dm with respect to the attribute A.

Step4: For each value ai of A Grow a branch from node N for the condition ai

• If Di is empty then attach a leaf labeled with the most common class in D.

• Else attach the node returned by MGI(Di)

Similar to other algorithms for building decision trees, in this algorithm the decision tree is constructed in a top-down recursive divide-and-conquer manner. At start, all the training tuples are at the root. The attributes are categorical. If an attribute is continuous valued, it is discretized in advance. Tuples are partitioned recursively based on selected attributes using its Gini ratio. Conditions for stopping partitioning are (a) all tuples for a given node belong to the same class, or (b) there are no samples left, or (c) there are no remaining attributes for further partitioning. In this case, majority voting is employed for classifying the leaf.

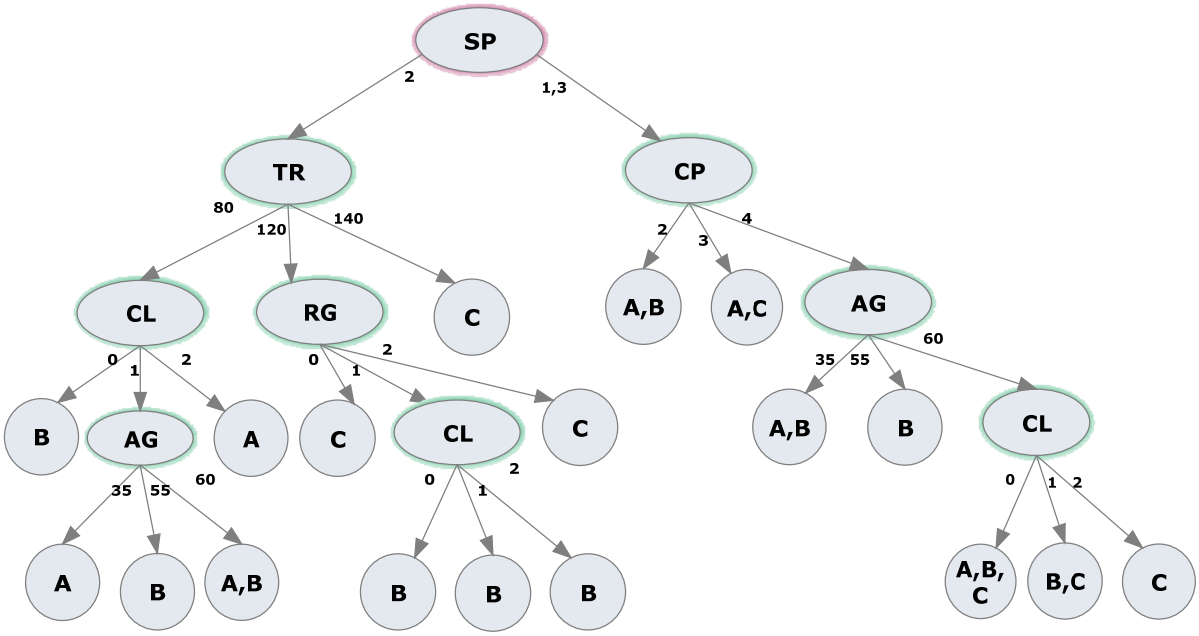
3.5. Methodology using heartdataset with four comparision Algorithms:

Heart data set has 100 tuples with 7 attributes.The label attribute refers to the presence of heart disease in the patient. It is valued from A (no presence) to C.To clean the data, data processing techniques are applied on the data.Then the cleaned data is divided into two parts trained data and test data.Trained data is 75% of the original data and it is used to calculate the classifier accuracy and Test data is 35% of the Original data and it is used to calculate the prediction accuracy.

First, Train dataset is processed to calculate accuracies for the four classification algorithms, ID3, C4.5, GINI INDEX and MODIFIED GINI INDEX. After estimating accuracies for the four comparing algorithms. Report is shown in the form of Graph the results show that the MGI algorithm has an overall better accuracy for heart data set. Moreover, the modified Gini index approach improved the accuracy in comparison with the traditional Gini index approach while C4.5 algorithm has a slightly worse accuracy in comparison with the ID3 algorithm.

Secondly, Test dataset is used to calculate the prediction accuracy for the four Comparing algorithms, the result is shown in the comparision graph of classifier accuracy and prediction accuracy. Moreover, the modified Gini index approach improved the accuracy in comparison with the traditional Gini index approach while C4.5 algorithm has a slightly worse accuracy in comparison with the ID3 algorithm.

C4.5 Algorithm Tree generated by calculating Infromation gain for each Attribute:-

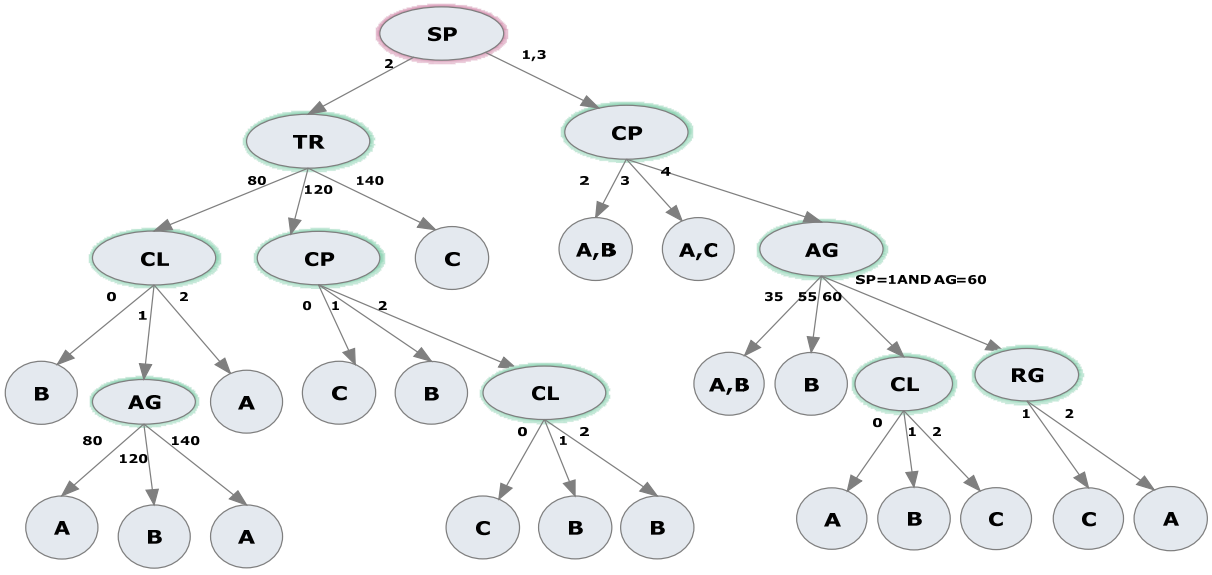


Total Leaf nodes are 25(Decreased when compared to Id3 algorithm)

Classifier Accuracy=72%

Prediction Accuracy=40%

Gini Index Tree is Genearted by calculating the splitting formulae for each Attribute:-

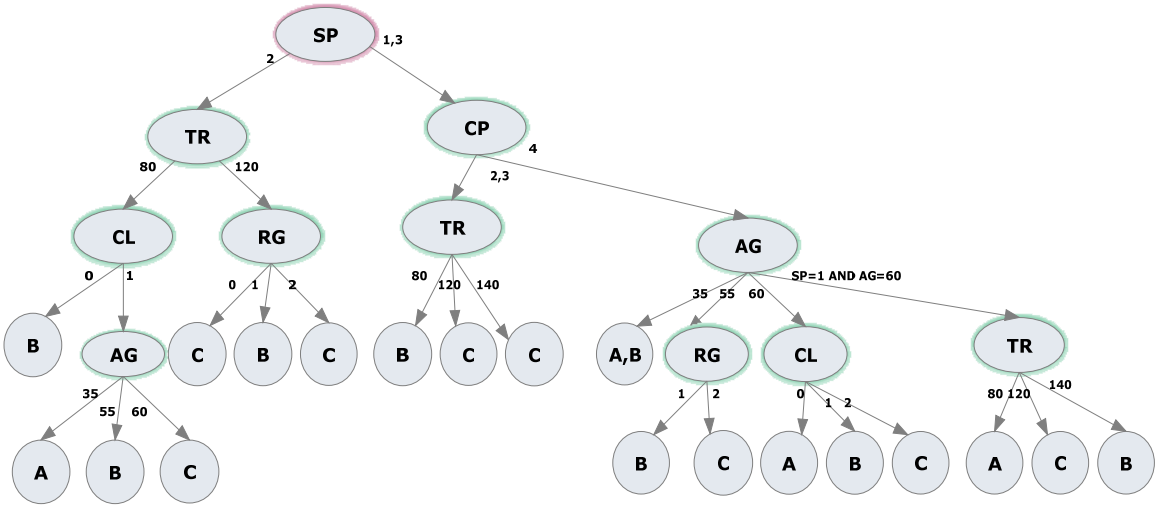


Total number of Leaf Nodes are 23(one leaf node decreased when comopared to C4.5 Tree)

Classifier Accuracy=74.66%

Prediction Accuracy=48%

Advanced Gini Index Tree is Generated by calculating the modified Spliting formulae:-



Total number of Leaf Nodes are 20(Nodes decreased when compared to all the Algorithms)

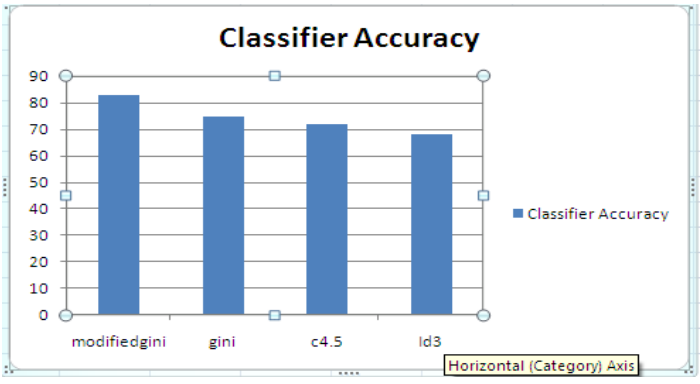
Classifer Accuracy=82.66%

Prediction Accuracy=60%

EXPERIMENTAL RESULTS

Classifier Accuracy Improvement (Experimental Accuracy):

To evaluate the system performances on Heart dataset collected from the UCI data repository. In order to evaluate the performance of the proposed Modified Gini-Index algorithm under different Classification Algorithms. In order to improve the accuracy of decision tree we have applied a techq. Of Spilting through ratios of Gini Index i.e..



The graphs shows that MGI has the overall better accuracy The comparisons among all the Classification Algorithms for calculating the Classifier Accuracy and Prediction Accuracy

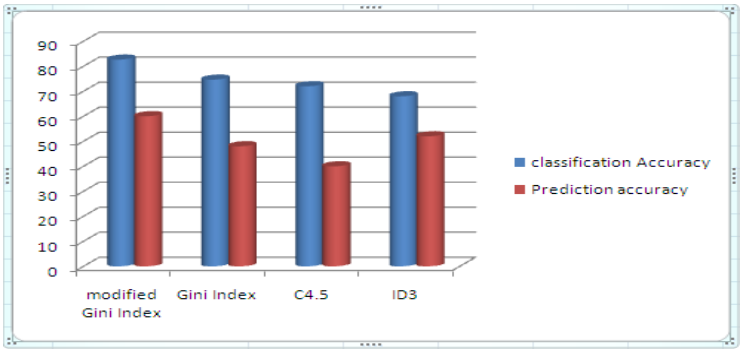
CALCULATION OF VARIOUS ACCURACIES:

METHOD	CLASSIFICATION ACCURACY	PREDICTION ACCURACY
ID3	51*100/75=68.0	13*100/25=52.0
C4.5	54*100/75=72	10*100/25=40.0
GINI-INDEX	56*100/75=74.66	12*100/25=48.0
ADVANCED GINI INDEX	62*100/75=82.666	15*100/25=60.0

Heart Dataset contains totally 100 attributes.
Trained Data consists of 75 tuples
Test data consists of 25 Tuples

Classificatio Accuracy and Prediction Accuracy is Calculated by using the formulae :

Classifier Accuracy
=i(total Tuples-unused)*100/(no of Attributes)
Total Tuples=75
Prediction Accuracy
=i(total tuples-unused)*100/(no of Attributes)
Total Tuples=25
Comparision of Classifier Accuracy and prediction in the form of Graph



Series1=Classifier Accuracy

Series2=Prediction Accuracy

CONCLUSION

I have proposed a tree - base approachs to analyzing multiple response using classification algorithms comparing with a modified decision tree method for classification to overcome the known problems for the Gini-based decision tree method, normalizing the Gini indexes by taking into account information about the splitting status of all attributes. Instead of using the Gini index for attribute selection as usual, we use ratios of Gini indexes and their splitting values in order to reduce the biases. Experimented with benchmark medical Heart Diseases dataset shows the modified decision tree method reacts differently with the heart dataset when compared too other known decision tree methods. The modified Gini index method performs well for some data bases but may not be the best for the others. Compared to previous multivariate decision tree methods that have limitations on the type of response and size of data the proposed method can analyze any type of multiple response by using this split formulae. Hence, it is called as"modified multivariate decision tree". A decision tree model that can handle numerical data attributes is developed to classify. It is a modified decision tree that can process all types of data attributes

References

[1] J. MacQueen, "Some methods for classification and analysis of mutlivariate observations," *Proc. 5th Berkeley Symp.Math. Statist, Prob.*, vol. 1, pp. 281–297, 1967.
[2] *Classification and Regression Trees*. Wadsworth International Group, 1984..
[3] W. J. Clancey, "Classification problem solving," in *Proceedingsof the National Conference on Artificial Intelligence* (R. J. Brachman, ed.), (Austin, Texas), pp. 49–55, William Kaufmann, Aug. 1984
4] Mining Medical Databases with Modified Gini Index Classification Quoc-Nam Tran Lamar University, U.S.A. qxtran@my.lamar.edu
[5] J. Quinlan, "Introduction to decision trees," *Machine Learning*, 1986.
[6] J. Quinlan, "Simplifying decision trees," *International Journal of Man-Machine Studies*, 1987.
[7] P. Pan, G. Swallow, and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels," May 2005, Internet
[8] U. Fayyad and K. Irani, "The attribute selection problem in decision tree generation," in *Proc. of AAAI-92*, pp. 104–110, 1992.
[9] P. Cheeseman and J. Stutz, "Bayesian classification (AUTOCLASS):Theory and results," in *Advances in Knowledge Discovery and Data Mining* (U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds.), 1995.