International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-8, June 2020

Classification Algorithm in Data Mining Based on Maximum Exponential Class Counts Technique

D. Mabuni

Abstract: ***A new split attribute measure for decision tree node split during decision tree creation is proposed. The new split measure consists of the sum of class counts of distinct values of categorical attributes in the dataset. Larger counts induce larger partitions and smaller trees there by favors to the determination of the best spit attribute. The new split attribute measure is termed as maximum exponential class counts (MECC). Experiment results obtained over several UCI machine learning categorical datasets predominantly indicate that the decision tree models created based on the proposed MECC node split attribute technique provides better classification accuracy results and smaller trees in size than the decision trees created using popular gain ratio, normalized gain ratio and gini-index measures. The experimental results are mainly focused on performing and analyzing the results from the node splitting measures alone.***

Keywords : Categorical Attributes, Categorical Datasets, Larger Counts And Larger Partitions, Maximum Exponential Class Counts (MECC).

I. INTRODUCTION

Decision trees are constructed in a top down, recursive, and divide and conquer methodology. Data splitting technique in decision tree creation is the most important step amongst all the steps of a decision tree creation and management. The fundamental goal of decision tree learning is how to produce compact decision trees with well improved generalization capabilities. Decision tree is a reliable, robust, benchmark, interpretable and efficient classification tool in machine learning. The divide and conquer strategy of a decision tree classifier creation is very efficient. A major advantage in exploring data is that a decision tree learning algorithm is scalable in terms of dataset size and the number of attribute dimensions. Classification accuracy increases as the size of the decision tree increases but large decision trees are less interpretable. Classification accuracy must be sacrificed to produce the compact and smaller sized decision trees. In the literature decision tree research persons are continuously investigating for finding the best data separation rules from the dataset for improving the performance, interpretability and comprehensibility features of decision tree classifiers. The most important and critical issue in the decision tree learning is the node data splitting criteria. In the recent years many persons have been proposed many improved decision tree algorithms with different perspectives. Decision trees are popular data classification models consisting of two types of nodes: internal (decision or branching) nodes and external (leaf) nodes. Each internal node represents a predictor attribute and each external node represents a class label. The measures Gain, Gain ratio, normalized gain ratio, Gini-index and miss classification error are some of the important split attribute techniques involving node impurity measures. Probably normalized gain ratio is considered to be the best data split technique in decision tree model creation. The most important disadvantage of a decision tree is that it does not consider other attributes in finding the best split attribute. Each attribute is treated as an independent attribute in determining the best split attribute.

Divide and conquer paradigm is the fundamental characteristic of the decision tree. There are two steps in the decision tree classifier creation. Decision tree is created in the first step from the given training dataset and it is tested in the second step using test instances. The accuracy of the decision tree is defined as the number of test instances that are correctly classified by it. The strength of the decision tree is its interpretability and comprehensibility. Sometimes pruning methods will be very much useful in reducing the size of the decision tree with sacrificed decision tree accuracy. The most popular decision tree node data separation rules remain the gain ratio, normalized gain ratio and the Gini-index but Gini-index results only binary splits that cause larger height decision trees. In decision tree learning the training data are divided into many subsets according to the values of the best split attribute selected dynamically. The intention is to produce larger sized subsets with greater and greater purity features. The decision tree construction algorithm proceeds recursively until all input attributes are exhausted or all instances belongs the same class or a pre specified threshold is attained. Decision tree construction algorithm proceeds with greedy approach. That is constructing the decision tree by creating one node at a time without any pre planned procedure. Big decision trees cause over fitting problems and small trees cause under fitting problems and both the problems must be balanced conveniently. Ensembles of classification and regression trees are mostly producing high classification and regression results.

Revised Manuscript Received on June 30, 2020.
***** Correspondence Author
D. Mabuni*, Department of Computer Science, Dravidian University, Kuppam, India. Email: mabuni.d@gmail.com

Decision trees are very good in prediction and computationally efficient both in training as well as in testing. The main requirement is that developed decision tree model must classify unknown or unseen or future test instances correctly. Pre pruning and post pruning are two special techniques useful for reducing the effect of over-fitting of decision trees. Futures of decision trees are: decision trees produce comparable classification accuracy results when compared with other methods, faster learning speed, and ability to use sql queries for accessing databases, and convertible to simple and easy understandable classification rules. Sometimes for any given dataset, decision tree model is constructed as a benchmark model for classification before applying the dataset on other machine learning techniques. Based on the bench mark results of the decision tree model with known parameters the same dataset is then applied on the other machine learning techniques.

II. RELATED WORK

In the decision tree literature number of node splitting rules is proposed by many research persons. The partial list of split measures is shown in TABLE-1.

TABLE-1 List of split attributes measures

Gain	Chi-square
Gain ratio	Twoing-rule
Normalized gain ratio	Minimum description length
Average gain ratio	Mean deviation measure
Gini-index measure	

Out of all these data split measures normalized gain ratio and Gini index rules are the popular and good in many cases but Gini-index produces binary splits resulting larger decision tree classifiers. So, normalized gain ratio is considered to be the best node data split technique in the literature of decision tree creation.

ID3 algorithm does not apply any pruning techniques. Continuous attributes and missing values are not handled by ID3. The C4.5 algorithm can handle missing values and continuous attributes in addition to the categorical attributes.

A. P. Bremner [1] thoroughly discussed and concluded that Gini-index measure always tries to separate largest class from all other classes in the dataset. B. Chandra et. al. [2] proposed a new splitting measure called distinct class based splitting measure in decision tree learning. Number of distinct classes plays a major role in the proposed new measure. This measure is based on the product of two terms. In the product, value of the first term increases as the total number of distinct classes in the partition increases and the second term decreases as the size of a class increases. J. Kent Martin [3] proposed several decision tree node split measures and then analyzed thoroughly and pointed out that the choice of split selection measure is not related or influences the accuracy of the decision tree model but it greatly affects the complexity, efficiency, and effectiveness of the pruning methods.

J. R. Quinlan [4] proposed ID3 decision tree algorithm based on information gain split measure but this measure is biased towards attributes having many distinct values and more over it supports only categorical splits. J. R. Quinlan [5] proposed C4.5 decision tree creation algorithm based on gain ratio split measure. This algorithm is one of the best data mining algorithms. It handles both categorical and continuous attributes. J. R. Quinlan [6] proposed C4.5 like decision tree algorithm which is an improved version of original C4.5 algorithm in terms of continuous attributes split measure evaluation. L. Breiman et al. [7] proposed a decision tree algorithm based on Gini-index split measure. It allows only binary splits resulting larger sizes of decision trees which ultimately leads to more and more computational complexity in decision tree learning.

R. Lopez De Mantaras [8] introduced a new attribute selection measure for ID3 like decision creation algorithms based on distance between data partitions. The proposed measure is not biased towards the attributes having the larger number of distinct values. S. Ruggieri [9] developed a new decision tree creation algorithm called EC4.5 which is an improved version of standard C4.5 algorithm. Syed Jawad Ali Shah and Qamruz Zaman [10] proposed a new split measure called mean deviation based measure for decision tree learning. After analyzing the experimental results they said that entropy measures are best suitable for balanced datasets whereas mean deviation measure and exponent based measures are best suitable for imbalanced datasets.

Sebastian Nowozin [11] identified that existing split measures are biased measures and they must be replaced with improved state of the art split measures. Wei-Yin Loh and Yu-Shan Shih [12] proposed split selection rule with negligible bias for decision tree generation. It produces binary splits and also applies direct stopping rule or efficient pruning technique for improving search efficiencies. Xinmeng Zhang and Shengyi Jiang [13] proposed a decision tree algorithm called mstree based on new data splitting criteria with maximum similarity determined with cluster similarity feature.

III. PROBLEM DEFINITION

Decision tree creation algorithm follows the standard top down approach starting from the root node to leaves. At each decision node data is split into subgroups based on some criteria of the attributes.

The best split attribute guides the data split process. Finding the best split attribute from among the set of potential attributes is the actual problem in decision tree creation. Many split data techniques are available but no one technique is good for all real applications.

Research persons are continuously trying to find the best split attribute. The problem is how to produce smaller decision trees with the best data separation rule in decision tree creation process.

IV. PROPOSED METHOD

A new categorical data split measure called maximum sum of exponential function class counts (MECC) is proposed in decision tree classifier generation. MECC method is based on maximum of sum of exponential function of class counts of distinct attribute values.

Sum of exponential function of class counts are computed for each attribute separately and then the attribute with the maximum sum is selected as the best split attribute and then the values of the best split attribute are used to separate the current node data into distinct number of partitions which are equal to the number of distinct categorical values of the attribute. This process is repeated recursively on the fly for each internal node of the decision tree until the newly created node becomes a leaf node. For easy understanding purpose a sample and hypothetical training dataset is given in TABLE-2 and another sample testing dataset is given in TABLE-3.

TABLE-2: LOAN_TRAINING DATASET

Age	Income	Profession	Surety	Loan
Child	Vrich	Govt	Yes	1
Child	Vrich	Govt	No	1
Child	Vrich	Private	Yes	1
Child	Vrich	Private	No	0
Child	Vrich	Agriculture	Yes	1
Child	Vrich	Agriculture	No	0
Child	Vrich	Business	Yes	1
Child	Vrich	Business	No	0
Child	Vrich	Unemp	Yes	1
Child	Vrich	Unemp	No	2
Child	Rich	Govt	Yes	1
Child	Rich	Govt	No	1
Child	Rich	Private	Yes	1
Child	Rich	Private	No	0
Child	Rich	Agriculture	Yes	1
Child	Rich	Agriculture	No	0
Child	Rich	Business	Yes	1
Child	Rich	Business	No	0
Child	Rich	Unemp	Yes	1
Child	Rich	Unemp	No	2
Child	Poor	Govt	Yes	1
Child	Poor	Govt	No	1
Child	Poor	Private	Yes	1
Child	Poor	Private	No	0
Child	Poor	Agriculture	Yes	1
Child	Poor	Agriculture	No	0
Child	Poor	Business	Yes	1
Child	Poor	Business	No	0
Child	Poor	Unemp	Yes	1
Child	Poor	Unemp	No	2
Child	Vpoor	Govt	Yes	1
Child	Vpoor	Govt	No	1
Child	Vpoor	Private	Yes	1
Child	Vpoor	Private	No	0
Child	Vpoor	Agriculture	Yes	1
Child	Vpoor	Agriculture	No	0
Child	Vpoor	Business	Yes	1
Child	Vpoor	Business	No	0
Child	Vpoor	Unemp	Yes	1
Child	Vpoor	Unemp	No	2
Young	Vrich	Govt	Yes	1
Young	Vrich	Govt	No	1
Young	Vrich	Private	Yes	1
Young	Vrich	Private	No	0
Young	Vrich	Agriculture	Yes	1
Young	Vrich	Agriculture	No	0
Young	Vrich	Business	Yes	1
Young	Vrich	Business	No	0
Young	Vrich	Unemp	Yes	1
Young	Vrich	Unemp	No	2
Young	Rich	Govt	Yes	1
Young	Rich	Govt	No	1
Young	Rich	Private	Yes	1
Young	Rich	Private	No	0
Young	Rich	Agriculture	Yes	1
Young	Rich	Agriculture	No	0
Young	Rich	Business	Yes	1
Young	Rich	Business	No	0
Young	Rich	Unemp	Yes	1
Young	Rich	Unemp	No	2
Young	Poor	Govt	Yes	1
Young	Poor	Govt	No	1
Young	Poor	Private	Yes	1
Young	Poor	Private	No	0
Young	Poor	Agriculture	Yes	1
Young	Poor	Agriculture	No	0
Young	Poor	Business	Yes	1
Young	Poor	Business	No	0
Young	Poor	Unemp	Yes	1
Young	Poor	Unemp	No	2
Young	Vpoor	Govt	Yes	1
Young	Vpoor	Govt	No	1
Young	Vpoor	Private	Yes	1
Young	Vpoor	Private	No	0
Young	Vpoor	Agriculture	Yes	1
Young	Vpoor	Agriculture	No	0
Young	Vpoor	Business	Yes	1
Young	Vpoor	Business	No	0
Young	Vpoor	Unemp	Yes	1
Young	Vpoor	Unemp	No	2
Old	Vrich	Govt	Yes	1
Old	Vrich	Govt	No	1
Old	Vrich	Private	Yes	1
Old	Vrich	Private	No	0
Old	Vrich	Agriculture	Yes	1
Old	Vrich	Agriculture	No	0
Old	Vrich	Business	Yes	1
Old	Vrich	Business	No	0
Old	Vrich	Unemp	Yes	1
Old	Vrich	Unemp	No	2
Old	Rich	Govt	Yes	1
Old	Rich	Govt	No	1
Old	Rich	Private	Yes	1
Old	Rich	Private	No	0
Old	Rich	Agriculture	Yes	1
Old	Rich	Agriculture	No	0
Old	Rich	Business	Yes	1
Old	Rich	Business	No	0
Old	Rich	Unemp	Yes	1
Old	Rich	Unemp	No	2
Old	Poor	Govt	Yes	1
Old	Poor	Govt	No	1
Old	Poor	Private	Yes	1
Old	Poor	Private	No	0

Old	Poor	Agriculture	Yes	1
Old	Poor	Agriculture	No	0
Old	Poor	Business	Yes	1
Old	Poor	Business	No	0
Old	Poor	Unemp	Yes	1
Old	Poor	Unemp	No	2
Old	Vpoor	Govt	Yes	1
Old	Vpoor	Govt	No	1
Old	Vpoor	Private	Yes	1
Old	Vpoor	Private	No	0
Old	Vpoor	Agriculture	Yes	1
Old	Vpoor	Agriculture	No	0
Old	Vpoor	Business	Yes	1
Old	Vpoor	Business	No	0
Old	Vpoor	Unemp	Yes	1
Old	Vpoor	Unemp	No	2

TABLE-3: LOAN_TESTING DATASET

Age	Income	Profession	Surety	Loan
Child	Vrich	Govt	No	1
Child	Vrich	Agriculture	Yes	1
Child	Vrich	Business	No	0
Child	Rich	Govt	Yes	1
Child	Rich	Private	No	0
Child	Rich	Business	Yes	1
Child	Rich	Unemp	No	2
Child	Poor	Private	Yes	1
Child	Poor	Agriculture	No	0
Child	Poor	Unemp	Yes	1
Child	Vpoor	Govt	No	1
Child	Vpoor	Agriculture	Yes	1
Child	Vpoor	Business	No	0
Young	Vrich	Govt	Yes	1
Young	Vrich	Private	No	0
Young	Vrich	Business	Yes	1
Young	Vrich	Unemp	No	2
Young	Rich	Private	Yes	1
Young	Rich	Agriculture	No	0
Young	Rich	Unemp	Yes	1
Young	Poor	Govt	No	1
Young	Poor	Agriculture	Yes	1
Young	Poor	Business	No	0
Young	Vpoor	Govt	Yes	1
Young	Vpoor	Private	No	0
Young	Vpoor	Business	Yes	1
Young	Vpoor	Unemp	No	2
Old	Vrich	Private	Yes	1
Old	Vrich	Agriculture	No	0
Old	Vrich	Unemp	Yes	1
Old	Rich	Govt	No	1
Old	Rich	Agriculture	Yes	1
Old	Rich	Business	No	0
Old	Poor	Govt	Yes	1
Old	Poor	Private	No	0
Old	Poor	Business	Yes	1
Old	Poor	Unemp	No	2
Old	Vpoor	Private	Yes	1
Old	Vpoor	Agriculture	No	0
Old	Vpoor	Unemp	Yes	1

Age attribute has 3 distinct categorical values, Income attribute has 4 distinct categorical values, Profession attribute has 5 distinct categorical values and Surety attribute has 2 distinct categorical values. The total number of instances thus formed is equal to 3 x 4 x 5 x 2 = 120. Out of total possible 120 instances, 80 instances are included in the LOAN_TRAINING_SET and 40 instances are included in the LOAN_TEST_SET.

The LOAN dataset is created based on two pre assumptions on the data values in the dataset. The first assumption is that the loan will be sanctioned for all those persons who will give correct surety certificate. The second assumption is that the loan will be sanctioned for all government employees only. That is, for receiving a loan amount the person must be either belongs to government employee category or must produce correct surety certificate.

A. Datasets Description

1) Breast Cancer Dataset

Breast cancer dataset is downloaded from the UCI machine learning repository. It consists of 286 instances which are divided into training set consisting of two third instances with size 191 and testing set consisting of one third instances with size 95. Breast cancer dataset is described with ten categorical attributes out of which nine are predictive attributes and tenth one is class label attribute with two class labels 0 and 1. Classes are

1) no-recurrence-events: 201 instances – class label - 0
2) recurrence-events: 85 instances – class label - 1

2) Car Dataset

Car Dataset is downloaded from the UCI machine learning repository. Car dataset consists of total 1728 records which are divided into three fourth training dataset with 1296 records and one fourth testing dataset with 432 records. Car data is described with seven categorical attributes and the seventh attribute is the class label with 4 distinct class labels denoted represented by 0, 1, 2, and 3.

3) Nursery Dataset

Nursery Dataset is downloaded from the UCI machine learning repository. Nursery Dataset consists of 12960 tuples. It is divided into three fourth training dataset with 9720 instances and one fourth testing dataset with 3240 instances. The dataset is described with nine attributes and ninth attribute is the class attribute. Five class labels are – not-recom, recommend, very-recom, priority, and spec-prior (special priority).

4) Primary Tumor Dataset

Primary tumor dataset is downloaded from the UCI machine learning repository. Primary Tumor dataset is divided into training dataset with 255 instances and testing dataset with 85 instances. This set is described with 18 attributes and 18th attribute is the class attribute and the remaining 17 are predictor attributes.

5) Lymphograhy Dataset

Lymphograhy dataset is downloaded from the UCI machine learning repository. Lymphograhy dataset is divided into training set with 119 instances and the testing set with 36 instances. Number of attributes in the dataset is 19. There are 18 predictor attributes and the last one is class attribute.

6) Hayes Roth Dataset
Hayes Roth dataset is divided into one training set with 132 instances and one testing set with 28 instances. It consists of 5 predictor attributes and 1 class attribute.
7) SPECT Dataset
It consists of heart data as training dataset with 80 instances and as testing dataset with 187 instances. Number of attributes is 22 predictor and 1 class attribute. Training dataset consists of 40 class 0 instances and 40 class 1 instances. Testing dataset consists of 15 class 0 instances and 172 class 1 instances
8) Balance Scale Dataset
It is divided into one training dataset and one testing dataset. The size of training dataset is 470 whereas the size of testing dataset is 156. It consists of 4 predictor attributes and one class attribute.
9) Loan Dataset
Loan dataset consists of 120 instances. It is divided into training dataset consisting of 80 instances and testing dataset consisting of 40 instances. There are three classes denoted by 0, 1, 2 respectively. Class label 1 means that the person is eligible for loan, class label 0 means the person is not eligible for loan and class label 2 means the person is in waiting list.

TABLE-4: Class counts of Age attribute

Age	Class-1	Class-2	Class-3
Child	12	24	4
Young	12	24	4
Old	12	24	4

Age = “Child” value class counts
Exp(12) = 162754.79141900392
Exp(24) = 2.648912212984347E10
Exp(4) = 54.598150033144236
Age = “Young” value class counts
Exp(12) = 162754.79141900392
Exp(24) = 2.648912212984347E10
Exp(4) = 54.598150033144236
Age = “Old” value class counts
Exp(12) = 162754.79141900392
Exp(24) = 2.648912212984347E10
Exp(4) = 54.598150033144236
MECC(Age) =
Exps(Age=Child) +Exps(Age=Young) + Exps(Age=Old)
MECC(Age) = 3[exp(12) + exp(24) + exp(4)]

= 3(162754.79141900392 + 2.648912212984347E10 +
54.598150033144236)
MECC(Age) = 7.946785481769913E10

TABLE-5: Class counts of Income attribute

Income	Class-1	Class-2	Class-3
Vrich	9	18	3
Rich	9	18	3
Poor	9	18	3
Vpoor	9	18	3

Income class counts are
Income = “Vrich” value class counts
Exp(9) = 8103.083927575384
Exp(18) = 6.565996913733051E7
Exp(3) = 20.085536923187668
Income = Rich class counts
Exp(9) = 8103.083927575384
Exp(18) = 6.565996913733051E7
Exp(3) = 20.085536923187668
Income = “Poor” value class counts
Exp(9) = 8103.083927575384
Exp(18) = 6.565996913733051E7
Exp(3) = 20.085536923187668
Income = “Vpoor” class counts
Exp(9) = 8103.083927575384
Exp(18) = 6.565996913733051E7
Exp(3) = 20.085536923187668
MECC(Income) =
Exps(Income=Vrich) +Exps(Income=Rich) +
Exps(Income=poor) + Exps(Income=Vpoor)
MECC(Income) = 4[exp(9) + exp(18) + exp(3)]

= 4[8103.083927575384 + 6.565996913733051E7 +
20.085536923187668]
MECC(Income) = 2.6267236922718E8

TABLE-6: Class counts of Profession attribute

Profession	Class-1	Class-2	Class-3
Agriculture	12	12	0
Business	12	12	0
Govt job	0	24	0
Private job	12	12	0
Unemployee	0	12	12

Profession attribute class counts are:
Profession = “Agriculture” value class counts
Exp(12) = 162754.79141900392
Exp(12) = 162754.79141900392
Exp(0) = 1.0
Profession = “Business” value class counts
Exp(12) = 162754.79141900392
Exp(12) = 162754.79141900392
Exp(0) = 1.0
Profession = “Govt job” value class counts
Exp(0) = 0
Exp(24) = 2.648912212984347E10
Exp(0) = 1.0
Profession = “Private job” value class counts
Exp(12) = 162754.79141900392
Exp(12) = 162754.79141900392
Exp(0) = 1.0
Profession = “Un-employee” value class counts
Exp(0) = 1.0
Exp(12) = 162754.79141900392
Exp(12) = 162754.79141900392
MECC(Profession) = 2.649042417417482E10

TABLE-7 Class counts of Surety attribute

Surety	Class-1	Class-2	Class-3
Yes	36	12	12
No	0	60	0

MECC(Surety) = Exp(36) + Exp(12) + Exp(12) + Exp(0) + Exp(60) + Exp(0)

= 4.311231547115195E15 + 162754.79141900392 + 162754.79141900392 + 1.0 + 1.1420073898156842E26 + 1.0
MECC (Surety) = 1.1420073898587966E26.

= Maximum of {MECC (Age), MECC (Income), MECC (Profession), MECC (Surety)}

= Maximum (7.946785481769913E10, 2.6267236922718E8, 2.649042417417482E10, 1.1420073898587966E26)
= 1.1420073898587966E26

At the beginning of the algorithm execution Surety attribute is selected as the best split attribute at the root node of the decision tree because MECC score of the surety attribute is maximum. Now the data in the root node is split into partitions based on the distinct categorical values of the surety attribute. Surety attribute has two distinct categorical values-Yes and No as a result the data is divided into two partitions one for surety = "No" and another for surety = "Yes". The second partition is 100 percent pure because all of its class labels belong to the same class. Therefore it is converted into the leaf node. Same process is applied recursively in each internal level of the decision tree. For surety = "No" partition, again remaining attributes are tested and determined Profession attribute as the best split attribute and as a result of this data is divided into five partitions corresponding to each distinct categorical value of the profession attribute and all these five partitions are converted into five leaf nodes of the Profession split attribute.

V. ALGORITHM

MECC algorithm is proposed for splitting data in decision tree node during decision tree creation. MECC is called from the main decision creation algorithm. The purpose of MECC is to find only the best split attribute for data partitioning based on the maximum value of exponential class counts of all distinct values of each of the categorical attribute. The decision tree is created recursively in top down fashion using divide and conquer methodology.

After execution of the proposed MECC split algorithm the output of the decision tree classifier is shown in Fig-1. Initially surety attribute is selected as the best split attribute and data is divided into partitions according to the values of surety.

When surety is "Yes" the partition is pure and when the surety is "No" once again the algorithm is executed and the profession is selected as the best split attribute and data is divided into partitions according to the distinct categorical values of the profession attribute.

The resulted decision tree is compact with height 3, two internal nodes and six leaf nodes. Also the rules are very easy to interpret with small rule set.

Algorithm MECC(R, D)
Input:
R root node of the decision tree
D is the training dataset
Output:
Decision tree classifier

1.for each attribute, i, in D find distinct categorical attribute values
2.for each categorical value, j, of the attribute find all class counts
3.find Exp(class count) for each class count
4. find sum of all Exp(class counts)
5. end for j
6. sum[i] = sum of Exp(class count) for all categorical Values
7.end for i
8. for each attribute k in D find
9. $maximum_k$ = find maximum of sum[k]
10.end for k
11. k^{th} attribute is selected as the best spit attribute
12. return the best split attribute to the decision tree creation main algorithm

Step-1: finds distinct categorical value for each attribute.

Step-2: for each distinct categorical value, v, find all class Counts
Step-3: for each class count finds Exp(class count)
Step-4-6: finds sum of all class counts for each attribute
Steps-8-10: find maximum Exp(class count) sum
Step-11: selects the best split attribute
Step-12: returns the best split attribute to the decision creation algorithm.

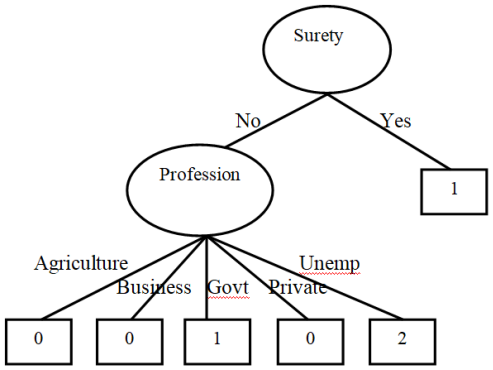

Fig-1 Decision tree for the data given in TABLE-1

The final decision tree created exactly resembles all the assumptions applied in the dataset while it was framing. Namely two rules are perfectly satisfied by the resulted decision tree classifier. It shows that the loan is sanctioned for all persons who are either government employees or correct surety certificate providers.

VI. EXPERIMENTS

Experiments are conducted by taking UCI machine learning categorical datasets. The results of experiments have shown that the proposed split data method is far better than the existing popular normalized gain ratio data split attribute measure. Accuracy of the decision tree classifier is dependent on the pruning threshold. Pre pruning threshold is applied in the present proposed method. Pruning results produce compact and simple decision tree classifiers and also reduces execution time.

TABLE-8: Experimental Results

Dataset Name	Training Data Size	Test Data Size	C4.5	MECC
Breast Cancer	192	95	61.053	**83.158**
Lymphography	112	36	27.777	**41.666**
Primary Tumor	254	85	55.294	**77.647**
Hayes-Roth	132	28	50	**71.429**
SPECT	80	187	58.824	**60.963**
Balance scale	469	156	45.513	**64.103**
Loan Data	80	40	100	**100**
Nursery Data	9719	3240	**86.358**	52.38
CAR	1296	432	**81.25**	76.62

Test datasets are taken as independent of training datasets to reduce the over fitting effect. Each dataset is conveniently divided into training set and testing set. Experiments are conducted by using both existing the best algorithm, C4.5 and the proposed algorithm MECC and the results are tabulated in TABLE-7. After careful observation of the results it is concluded that the proposed split attribute method, MECC, is superior in cases of seven datasets and the existing method is better in cases of two datasets.

VII. CONCLUSION

A new split rule called MECC is proposed in this paper for splitting node data in the decision tree creation. This rule is based on the maximum value of sum of exponential function class counts of distinct categorical values of attributes. The attribute with maximum MECC count is selected as the best split attribute. In the future this function will be extended with normalized class counts of distinct categorical values of attributes. Normalized measures are required to convert larger values into smaller values in order to make the computation process easier. In the future more number of UCI machine learning datasets will be used in experimentation.

REFERENCES

1. A. P. Bremner, "Localised Splitting Criterion for Classification and Regression Trees", Ph.D thesis, Murdoch University, 2004.
2. B. Chandra, RaviKothari, and PallathPaul, "A new node splitting measure for decision tree construction", ELSEVIER, Pattern Recognition 43 (2010) 2725–2731, Pattern Recognition,
3. J. KENT MARTIN, "An Exact Probability Metric for Decision TreeSplitting and Stopping", Machine Learning, 28, 257–291 (1997)1997 Kluwer Academic Publishers. Manufactured in The Netherlands.
4. J. R. Quinlan, Induction of decision trees. Machine Learning, vol.1,pp. 81-106, 1986.
5. J. R. Quinlan, C4.5: Programs for machine learning. 1st ed.San Mateo, CA: Morgan Kaufmann, 1993.
6. J. R. Quinlan, Improved use of continuous attributes in C4.5, Journal of Articial Intelligence Research, vol.4, pp.77-90, 1996.
7. L. Breiman, J. Friedman, R. Olsen, C. Stone, Classification and Regression Trees, Wadsworth International,1984.
8. R. Lopez De Mantaras, "A distance based attribute selection measure for decision tree induction", Machine Learning, 6, 81-92 (1991) © 1991 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
9. S. Ruggieri, Effcient C4.5. IEEE Transactions on Knowledge and Data Engineering,vol.14, pp.438-444, 2002.
10. Syed Jawad Ali Shah and Qamruz Zaman, "A Mean Deviation Based Splitting Criterion for Classification Tree"
11. Sebastian Nowozin, "Improved Information Gain Estimates for Decision Tree Induction", Appearing in Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK, 2012. Copyright 2012 by the author(s)/owner(s).
12. Wei-Yin Loh and Yu-Shan Shih, "SPLIT SELECTION METHODS FOR CLASSIFICATION TREES", Statistica Sinica 7(1997), 815-840
13. Xinmeng Zhang and Shengyi Jiang, "A Splitting Criteria Based on Similarity in Decision Tree Learning", JOURNAL OF SOFTWARE, VOL. 7, NO. 8, AUGUST 2012, © 2012 ACADEMY PUBLISHER doi:10.4304/jsw.7.8.1775-1782

AUTHORS PROFILE

D. Mabuni, completed M.Sc. (Computer Science), MCA and M.Phil. (Computer Science). Currently working as Assistant Professor in the Department of Computer Science at Dravidian University, Kuppam, Andhra Pradesh, India. My interested research areas are Data Mining, Databases, and User Interfaces.