A New Aggregated Attribute Values Match Technique for Improving the Quality of Probability Estimated Decision Trees

D. Mabuni

***Abstract:** Probability estimations of decision trees may not be useful directly because their poor probability estimations but the best probability estimations are desired in many useful applications. Many techniques have been proposed for obtaining good probability estimations of decision trees. Two such optical techniques are identified and the first one is single tree based aggregation of mismatched attribute values of instances. The second one is bagging technique but it is costly and less comprehensible. So, in this paper a single aggregated probability estimation decision tree model technique is proposed for improving the performance of probability estimations of decision trees and the performance of new technique is evaluated using area under the curve (AUC) evaluation technique. The proposed technique computes aggregate scores based on matched attribute values of test tuples.*

***Keywords** : Aggregate scores, bagging technique, mismatched and matched attribute values, poor probability estimations.*

I. INTRODUCTION

Decision trees are the most valuable and useful tools in data mining as well as in machine learning in the present day to day useful state-of-the-art data analysis and data classification techniques. Obtaining high data classification accuracy is the fundamental property of this tool. Usually the height of the tree will be more. Height of the tree will be reduced to some extent by sacrificing the classification accuracy. Out of many machine learning applications, the decision tree learning is considered to be the best data classification technique. Decision trees are very good for data classification but they have strong limitation in providing very good results of probability estimations. Decision tree algorithm follows divide-and-conquer principle. The decision tree created may produce the best classification results but may not always produce the best probability based rankings of test tuples. Probability estimated trees are the wonderful and attractive features of normal decision tree induction. In normal probability estimation trees, in each leaf, probability is available for each class. Different researchers have proposed different quality measures of probability estimation trees. Pruning generally always does not improve the quality measure of the probability estimates. Area under the curve (AUC) measure is predominantly using for evaluating the quality of the leaf probability estimations of the probability estimation decision trees in terms of ranks of test tuples. That is, AUC measure is a good measure for comparing probability estimated rankings. The goal of the present study is how to improve the probability based tuple rankings measured by AUC in decision trees. The best algorithm is one which produces different ranks for different test instances even though many instances fall into the same leaf node. Algorithms must be independent of tree size.

Many researchers have shown that the results produced by bagging are far better than other methods. Even though bagging is better than all the algorithms in terms of AUC scores it has two main disadvantages-high computational costs, less comprehensibility and less interpretability. One must apply trade off conditions such as computational cost, comprehensibility, execution time, quality of the results before going to select the best algorithm. Normally single decision tree is sufficient in many real world cases except under extraordinary situations.

Different smoothing techniques are proposed for improving the quality measures of the probability estimations of decision trees. Some of these quality measure improvement techniques are:

1. Frequency based estimation in leaves
2. Laplace correction for smoothing leaves
3. m-estimate smoothing
4. m-branch smoothing
5. ensemble of decision trees

Laplace technique is not good for finding probability estimates of decision trees when datasets are unbalanced but it improves probability estimation up to certain extent. Sometimes ensemble techniques are very useful for smoothing the decision tree probability estimations of the leaves. In many cases ranking order of tuple instances is far better than categorical data predictions. A probability estimation decision tree (PEDT) is a decision tree such that the probability score for each class in each leaf of the decision tree is available. Several efficient and effective techniques have been proposed for constructing more accurate probability estimation decision trees. Certainly there is a need to improve the performance of the probabilistic estimation of decision trees. One must investigate for obtaining standard quality measures for evaluating the probability estimation decision trees (PEDTs). Currently new techniques for leaf probability estimation are being proposed by many researchers in different disciplines and in different domains.

Revised Manuscript Received on May 30, 2020.

***** Correspondence Author
D. Mabuni*, Department of Computer Science, Dravidian University, Kuppam, India. Email: mabuni.d@gmail.com

In addition to the classification accuracy there is a demand to find other measures such as reliability, probability, high quality results, and likely hood and so on. Many different types of node splitting procedures are proposed for improving the leaf probability estimations rather than improving the classification accuracy. PEDTs need different types of node splitting measures for improving quality. Node splitting criterion plays an important role in the process of obtaining the best quality probability estimation decision trees (PEDTs). Pruning is advisable to apply only in certain cases but without sacrificing the quality of the PEDTs.

Decision trees are very good in giving classification accuracy but very poor in giving probability estimations of test tuples. Assigning a rank rather than assigning a class label of a test tuple is convenient in many applications. Probabilistic estimates or ranking ordering of decision trees is useful for reducing the expenses to be incurred on conducting further tests with the help of specified thresholds. A probability estimated decision tree is said to be well designed or well calibrated if the difference between the predicted probability and empirical probability is very close to zero when number of predictions increases to very large.

Probability estimated decision trees must have the features of interpretability, comprehensibility, scalability in terms of attribute dimensionality and the size of datasets, efficiency, accuracy, ease of handling complexity and so on. Several techniques have been proposed for improving the probability based rankings of decision trees but all those techniques have not been evaluated experimentally as well as systematically. One of the most important applications of probability estimated decision tree applications is speech recognition. The fraud detection system must produce ordered rankings of the frauds detected in the bank. The number of instances in the leaves must be large enough so that the probabilities estimated are extremely reliable. The goal of the present study is to create a learned tree model that can rank the given test tuples using probability of the class memberships in the leaves of the decision trees. Alina Beygelzimer et. al. [1] have efficiently estimated the conditional probability of the class label for a given tuple instance with time complexity O(log n) where n is the number of class labels.

ROC curves (AUC) are generated for evaluating the quality of rankings of the probability estimated decision trees. AUC measure is becoming a standard technique for evaluating the performance of the probability estimators and ordered rankings of probability decision tree models. Bagging is one technique for improving the quality of the probability estimates. The computational cost of implementing bagging is very high for decision tree probability estimates. So, in many cases a single probability estimate is sufficient for majority of the real-world applications with sacrificed quality. Generally, conventional decision tree learning algorithms will give poor performance results of probability estimates.

Many researchers have observed that bagging improves effectively more for probability estimates than the application of bagging for decision tree classification accuracy. Andrew J. Sage et. al. [2] empirically studied the performance results of aggregation approaches with respect to probability estimation of classification and regression. They found that aggregation effect is very small in regression but very high in probability estimation of classification problems. In many applications there is a need to construct good probability estimates rather than constructing a decision tree for good classification accuracy.

II. RELATED WORK

Applying data mining techniques on many live applications for getting useful results is increasing day by day. Some researchers have introduced new methods for finding quality measures of probability estimations of decision trees in terms of leaves instead of rankings of test tuples. In the literature tremendous efforts have been applied for increasing the area under the curve (AUC) quality measures of probability estimations of decision trees. Cesar Ferri et al. [3] have studied several issues for improving the performance results of probabilistic estimation of decision trees and they derived a new probability smoothing technique (M-branch smoothing technique) which considers probability distribution details of all the nodes from the root to leaf paths. They also proposed a new technique for tree node splitting with the aim of improving probability estimates.

C. Ferri et. al. [4] studied concepts and techniques for improving the performance results of probability estimation trees. They proposed a new technique for node splitting during tree construction with the goal of improving performances of probability estimation trees. Charless X. Ling and Robert J. Yan. [5] proposed a new algorithm for increasing the quality of the single probability estimation decision tree. The new algorithm estimates aggregate of probabilities from all the leaf nodes of the probability estimation decision tree instead of estimating the probability of the leaf node where the test tuple actually falls. Authors experimentally verified that proposed algorithm in terms of AUC score outperforms all the existing algorithms except bagging method. Cheng Zhang and Frederick A. Matsen, [6] devised a new technique for probability estimation based on Bayesian network working principle and showed that Bayesian networking models can easily be extended for finding probability estimations of other many models. Dragos and D. Margineantu, [7] proposed a new algorithm called bagged lazy option trees (B-LOTs) for constructing decision trees and these trees are compared with bagged probability estimation trees (B-PETs) and experimentally proved that B-LOTs produce more accurate probability estimates than B-PETs. Foster Provost and Pedro Domingos. [8] have studied several techniques useful for improving the probability based rankings of probability estimates of decision trees. Henrik Bostrom. [9] experimentally verified that a small forest of ten probability estimated decision trees has given a higher score of AUC than classification trees of forest with same size and also observed that performance decreases as the size of the forest increases. Han Liang et. al. [10] empirically studied traditional decision tree models for probability estimation by using conditional log likely hood measure. Decision tree models are compared with other learning models through experiments. The experimental results reveal that conditional log likely hood based decision tree model C4.4 is the best model in many applications but its performance is slightly lesser than AUC in some applications. In many applications probability estimations are very important rather than the classification accuracy.

In the case of majority real-world applications accurate probability estimation learning models are compulsory to use especially in special fields such as defense, medical, research, agriculture and so on. Isabelle Alvarez et. al. [11] proposed a new technique for smooth class probability estimations in decision trees by using decision boundaries in the decision trees.

Khan Z et. al. [12] proposed an ensemble technique consisting of optimal decision trees in terms of their predictive performance. Initially all the trees are ordered in predictive performance value and then trees one by one are added to form a forest of trees and a tree is added only when the predictive performance of the forest increases. Also random forests of trees and probability estimations are compared using Brier score performance measure. Kun Zhang and Bill P. Buckless. [13] said that don't use poor posterior probability decision tree estimators in applications and they studied various types of probability estimation tree algorithms to overcome the problem. Nitesh V. Chawla and David A. Cieslak. [14] have evaluated the quality of the probability estimates by using different types of loss measures and also studied the relationships between the quality of the probability estimates and the results of the rank ordering results of test instances.

Peter Flach and Edson Takashi Matsubara [15] proposed a lexicographic ranker algorithm for assigning ranks to the test instances based on values of attributes in the training dataset. Also verified that odds ratio to rank the attribute values is very close to the naïve Bayesian classifier. Rivaka an Haimonti Dutta. [16] have proposed a new algorithm for improving the performance of the ranked probability estimations of decision trees also they have proposed a new technique for node splitting procedure for improving ranking order of probability estimates.

III. PROBLEM DEFINITION

Crisp data classification results produced by decision trees are not always useful or sometimes such results are not required in many types of applications. In frequency based leaf probability estimations of decision trees a group of tuples with different values of attributes will be assigned to the same class. Strictly speaking this assignment technique is incorrect or irrelevant. When tuples differ in some of the attribute values, certainly there must be difference in their predictions also. This desired result is achieved by improving the prediction quality of the test instances in terms of their ranking or probability estimation order using smoothing techniques. In this paper, a new technique is proposed for increasing performance quality of probability estimations of decision trees and the results are evaluated using AUC method.

IV. PROBABILITY ESTIMATED DECISION TREES (PEDTs)

When class probabilities are computed in the leaf nodes of a decision tree then the tree is called probability estimated decision tree (PEDT). Generally, PEDTs are very poor in probability estimation. Hence, high quality and standard measures are needed for improving and evaluating the probability estimation models. Area under the curve (AUC) is one such technique frequently used for evaluating the quality of probability estimated decision trees. AUC measure is defined for two classes as well as for multi classes. AUC for two classes is $E = \sum_{i<j} A(i,j)$ and

AUC for multi class is $M = \frac{2}{c(c-1)} \sum_{i<j} A(i,j)$

Where c is the number of classes in the dataset and i means ith class and j means jth class and

$$A(i,j) = \frac{S - \frac{n_0(n_0+1)}{2}}{n_0 * n_1}$$

A(i, j) means AUC score for the two classes i and j.

n_0 is the number of positive classes

n_1 is the number of negative classes

For multi classes first compute AUC scores pairs of classes and then sum all these pair wise scores to give up a single multi class AUC score.

Different techniques have been proposed in the literature for improving the quality measures of the probability estimations of the decision trees. Frequency based estimation, Laplace correction, m-estimation, aggregated score of miss matched values of attributes. Existing miss matched values of attributes technique is considered to be the best technique for probability estimation in decision trees.

A. Existing Technique

Among all the probability estimation of decision trees the one which uses aggregate class membership probabilities of all the leaf nodes is considered to be the best technique for improving the quality of the PEDTs. Authors X. Ling and Robert J. Yan [15] proposed one of the best methods for probability estimation of decision trees. This method uses a single decision tree. Bagging method is the best method for finding probability estimations of decision trees; it uses multiple decision trees with high computational time complexity but gives more accurate results. Aggregate class membership probability of the tree is estimated based on the assumption that the randomly selected test tuple may fall in any of the leaf nodes. Aggregate measures in all the cases will give better results. So, aggregate probability measure formula is

$$P(c/x) = \frac{\sum_{i=1}^{n} P_i(c) * s^k}{\sum s^k} \qquad (1)$$

Where the value $P(c/x)$ is the aggregate class membership probability of the class c for the randomly selected test instance and $P_i(c)$ is the class c membership probability in the leaf node i and s is the confusion factor with possible values 0.1, 0.2, 0.3, 0.4, 0.5 and so on and k is the number of miss matched values of the attributes along the path from root node of the tree to the respective leaf node. Existing algorithm will assign different probabilities for the different test tuples instead of giving same probability for all the test tuples falling in the same leaf node of the decision tree. Existing algorithm improves the quality of PEDTs which is experimentally verified by computing the AUC scores of PEDTs. AUC scores are specifically used for evaluating the performances of PEDTSs.

B. Proposed Technique

A new technique is proposed for probability correction or estimation in decision trees.

It is an aggregate probability score evaluation method based on the matched values of attributes along the root node to the respective leaf paths. Once again the assumed assumption is that there is a possibility to fall a random test instance in any one of the leaf nodes of a decision tree. Note that the main difference between the existing and proposed techniques is that existing technique uses miss match values of attributes along the path from root to leaf whereas the proposed technique uses matched values of attributes along the path from root node to leaf node. The aggregate probability is computed by using the formula for each test instance.

$$P(c/x) = \sum_{i=1}^{n} P_i(c) * P_{i+1}(c) * P_{i+2}(c) * m_i \qquad (2)$$

Where P(c/x) is the aggregate class membership probability of the class c for the test instance x, and Pi(c), Pi+1(c) and Pi+2(c) are the respective probabilities of the ith, (i+1)th and (i+2)th leaf nodes of the corresponding probability estimation decision tree. Leaf node probabilities of all the leaf nodes in the tree are modified using the proposed technique and then the performances of both existing and proposed techniques are evaluated using AUC scores. The proposed technique has yielded superior performance results in more than 95% of the experiments and it is robust with respect to the tree parameters.

V. ALGORITHM

```
Algorithm Performance_EvaluationPEDT
1.create decision tree for the given dataset
2.compute class membership probabilities of the leaf
  nodes of the decision tree
3.find modified probabilities of the probability estimated
  decision tree using the formula
  $P(c/x) = \frac{\sum_{i=1}^{n} P_i(c) * s^k}{\sum s^k}$
4.compute AUC score for the new probabilities of the leaf
  nodes of the existing technique based for the modified
  tree
5.find modified probabilities of the original probability
  estimated decision tree using the formula
  $P(c/x) = \sum_{i=1}^{n} P_i(c) * P_{i+1}(c) * P_{i+2}(c) * m_i$
6.compute AUC score for the new probabilities of the leaf
  nodes of the proposed technique based modified tree

7.compare AUC scores of both the existing and proposed
  techniques.
8.output the best probability estimation decision tree
  whose AUC score is maximum.
```

A. Algorithm Explanation

Line-1 creates a decision tree model for the given dataset

Line-2 for each leaf in the tree class membership probabilities are created

Line-3 leaves probabilities are modified according to the aggregate probability scores for each test tuple instance using existing technique

Line-4 compute AUC score for the modified probabilities using existing technique

Line-5 for each leaf in the original tree class membership probabilities are computed

Line-6 leaves probabilities are modified according to the aggregate probability scores for each test tuple instance using proposed technique

Line-7 compute AUC score for the modified probabilities using proposed technique

Probability estimated decision tree for a particular dataset is shown in the Figure-1. The tree contains eight leaf nodes numbered as L1, L2, L3, L4, L5, L6, L7, and L8. For easy understanding purpose internal nodes are represented with circles and external nodes are represented with rectangles containing class membership probabilities.

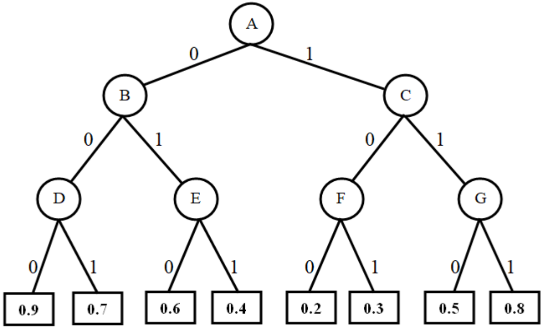

Figure-1 Sample Probability Estimation Decision Tree

T1, T2, T3, T4, and T5 are sample probability estimated decision trees and L1, L2, L3, L4, and L5 are respective leaf nodes of the corresponding probability estimation decision trees. Five probability estimated decision trees are shown in the TABLE-1.

TABLE-1 Probability probabilities estimation decision trees and their leaf

	L1	L2	L3	L4	L5	L6	L7	L8
T1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
T2	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
T3	0.1	0.3	0.5	0.7	0.8	0.6	0.4	0.2
T4	0.1	0.2	0.3	0.4	0.4	0.3	0.2	0.1
T5	0.9	0.8	0.7	0.6	0.6	0.7	0.8	0.9

AUC scores for both existing and proposed methods are computed for all the five probabilities estimated decision trees and the results are tabulated in the TABLE-2. Proposed method is very much better than the existing technique.

TABLE-2 Comparison of AUC scores of existing and proposed methods

Tree Name	Existing Method AUC score	Proposed Method AUC score
T1	0.791667	1.0
T2	0.208333	0.416667
T3	0.416667	0.5
T4	0.388889	0.5
T5	0.541667	0.875

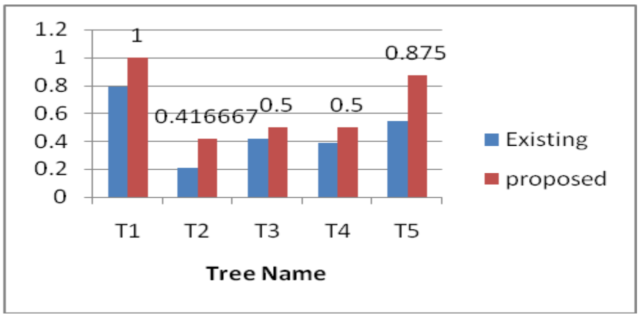

Figure-2 AUC scores comparison between existing and proposed methods

AUC score comparison results of both existing and proposed methods are graphically shown in Figure-2. X-axis represents different probability estimated decision trees and Y-axis represents AUC scores of trees. In all the five cases proposed attribute values matched technique is superior to the existing technique of miss matched values of attributes in terms of AUC scores. Many experiments are conducted by taking different types of probability estimation decision trees. In all the experiments conducted more than ninety percent of the cases proposed technique has produced better AUC performance evaluation results than the existing technique.

VI. EXPERIMENTS

In this paper the main goal is not to create an efficient probability estimation decision tree instead the goal is how to improve the quality of the probability estimation decision trees. AUC scores are used for evaluating qualities of the probability estimation decision trees. Quality increases as the AUC score of the probability estimation decision tree increases. A single tree based new technique is proposed in this paper for increasing the quality of the probability estimation decision trees. Existing technique works on the principle of miss matched attribute values whereas the proposed technique works on the principle of matched attribute values of test tuples. Extensive experiments are conducted by taking different decision trees.

Test dataset used for evaluating quality of probability estimated decision trees is shown in the TABLE-3. Class label “0” represents first class, class label “1” represents second class and class label “2” represents third class.

TABLE-3 Test Dataset for Evaluation

Test Instance Id	A	B	C	Class label
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	2
8	1	1	1	2

For testing purpose, the test instance (0, 0, 0) is applied along the path once for each leaf node of the probability estimation decision tree shown in the Figure-1. Newly computed or modified probability of the leaf node, L_i , is called the aggregated probability, which is computed by adding all the newly computed leaf probabilities with respect to the test instance (0, 0, 0) using the existing technique based equation-1.

Test instance-1 = (0, 0, 0)

Original probabilities of all leaf nodes are

(0.9, 0.7, 0.6, 0.4, 0.2, 0.3, 0.5, 0.8)

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-1 = 0

Existing formula result =0.9 * $(0.3)^0$ = 0.9

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-2 = 1

Existing formula result =0.7 * $(0.3)^1$ = 0.21

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-3 = 1

Existing formula result =0.6 * $(0.3)^1$ = 0.18

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-4 = 2

Existing formula result =0.4 * $(0.3)^2$ = 0.036

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-5 = 1

Existing formula result =0.2 * $(0.3)^1$ = 0.06

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-6 = 2

Existing formula result =0.3 * $(0.3)^2$ = 0.027

For the test instance-1 (0, 0, 0), number of mismatches of values of attributes in path-7= 2

Existing formula result =0.5 * $(0.3)^2$ = 0.045

For the test instance-1 (0, 0, 0), number of miss matches of values of attributes in path-8 = 3

Existing formula result =0.8 * $(0.3)^3$ = 0.0216

Aggregate for test tuple (0, 0, 0) = 0.9 + 0.21 + 0.18 + 0.036 + 0.06 + 0.027 + 0.045 + 0.0216 = 1.4796/8 = 0.18495

Similarly aggregate probability values for other test instances are computed.

Aggregate for test tuple (0, 0, 1) = 0.27 + 0.7 + 0.054 + 0.12 + 0.018 + 0.09 + 0.0135 + 0.072 = 1.3375/8 = 0.167188

Aggregate for test tuple (0, 1, 0) = 0.27 + 0.063 + 0.6 + 0.12 + 0.018 + 0.0081 + 0.15 + 0.072 = 1.3011/8 = 0.162638

Aggregate for test tuple (0, 1, 1) = 0.081 + 0.21 + 0.18 + 0.4 + 0.0054 + 0.027 + 0.045 + 0.24 = 1.1884/8 = 0.14855

Aggregate for test tuple (1, 0, 0) = 0.27 + 0.063 + 0.054 + 0.0108 + 0.2 + 0.09 + 0.15 + 0.072 = 0.9098/8 = 0.113725

Aggregate for test tuple (1, 0, 1) = 0.081 + 0.21 + 0.0162 + 0.036 + 0.06 + 0.3 + 0.045 + 0.24 = 0.9882/8 = 0.123525

Aggregate for test tuple (1, 1, 0) = 0.081 + 0.0189 + 0.18 + 0.036 + 0.6 + 0.27 + 0.5 + 0.24 = 1.1429/8 = 0.142863

Aggregate for test tuple (1, 1, 1) = 0.0243 + 0.063 + 0.054 + 0.12 + 0.018 + 0.09 + 0.15 + 0.8 = 1.3193/8 = 0.164913

TABLE-4 Existing technique based AUC computations for 0 and 1 classes

Aggregate probability	Class label	Rank
0.18495	0	6
0.167188	0	5
0.162638	0	4
0.14855	1	3
0.113725	0	1
0.123525	1	2

AUC score between class labels 1 (negative) and 2 (positive) is = AUC(0,1)

$$A(0,1) = \frac{S - \frac{n_0(n_0+1)}{2}}{n_0 * n_1}$$

$$A(0,1) = \frac{(3+2) - \frac{(2*3)}{2}}{2*4}$$

A(0,1) = (5-3)/8 = 2/8 = 0.25

TABLE-5 Existing technique based AUC computations for 0 and 2 classes

Aggregate probability	Class label	Rank
0.18495	0	5
0.167188	0	4
0.162638	0	2
0.142863	2	1
0.164913	2	3

$$A(0,2) = \frac{(1+3) - \frac{(2*3)}{2}}{2*3}$$

A(0, 2) = (4 – 3)/6 = 1/6 = 0.166667

TABLE-6 Existing technique based AUC computations for 1 and 2 classes

Aggregate probability	Class label	Rank
0.14855	2	4
0.113725	1	1
0.123525	2	2
0.142863	2	3
0.164913	2	5

$$A(1,2) = \frac{(4+2+3+5) - \frac{(4*5)}{2}}{4*1}$$

A(1, 2) = (14 – 10)/4 = 4/4 = 1

AUC for multi class is $M = \frac{2}{c(c-1)} \sum_{i<j} A(i,j)$

$$M = \frac{2}{3(3-1)} * [A(0,1) + A(0,2) + A(1,2)]$$

$$M = \frac{2}{3*2} * (0.25 + 0.166667 + 1)$$

M = 1.416667/3 = 0.472222

Similarly for the proposed technique the computation values are shown.

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-1 = 3

Proposed formula result =0.9 * 0.7 * 0.6 *3 =1.134

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-2 = 2

Proposed formula result = 0.7 * 0.6 * 0.4 * 2 = 0.336

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-3 = 2

Existing formula result =0.6 * 0.4 * 0.2 * 2 = 0.096

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-4 = 1

Proposed formula result =0.4 * 0.2 * 0.3 * 1 = 0.024

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-5 = 2

Proposed formula result =0.2 * 0.3 * 0.5 * 2 = 0.06

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-6 = 1

Proposed formula result =0.3 * 0.5*0.8*1 = 0.12

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-7= 1

Proposed formula result =0.5 * 0.8 * 1 * 1 = 0.4

For the test instance-1 (0, 0, 0), number of matches of values of attributes in path-8 = 0

Proposed formula result =0.8 * 1 * 1 * 0 = 0.0

Aggregate for test tuple (0, 0, 0) = 1.134 + 0.336 + 0.096 + 0.024 + 0.06 + 0.12 + 0.4 + 0 = 2.17/8 = 0.27125

Similarly aggregate probability values for other test instances are computed.

Aggregate for test tuple (0, 0, 1) = 0.756 + 0.504 + 0.048 + 0.048 + 0.03 + 0.24 + 0 + 0.512 = 2.14/8 = 0.26725

Aggregate for test tuple (0, 1, 0) = 0.756 + 0.168 + 0.144 + 0.048 + 0.03 + 0 + 0.8 + 0.512 = 2.458/8 = 0.30725

Aggregate for test tuple (0, 1, 1) = 0.378 + 0.336 + 0.096 + 0.072 + 0 + 0.12+ 0.5 + 0.1024 = 2.426/8 = 0.30325

Aggregate for test tuple (1, 0, 0) = 0.14288 + 0.056448 + 0.004608 + 0.001728 + 0 + 0.0144 + 0.16 + 0.524288 = 0.904356/8 = 0.113044

Aggregate for test tuple (1, 0, 1) = 0.378 + 0.336 + 0 + 0.024 + 0.06 + 0.36 + 0.04 + 1.024 = 2.6/8 = 0.32275

Aggregate for test tuple (1, 1, 0) = 0.378 + 0 + 0.096 + 0.024 + 0.6 + 0.12 + 1.2 + 1.024 = 2.902/8 = 0.36275

Aggregate for test tuple (1, 1, 1) = 0 + 0.0168 + 0.048 + 0.048 + 0.03 + 0.24 + 0.8 + 1.536 = 2.87/8 = 0.35875

TABLE-7 Proposed technique based AUC computations for 0 and 1 classes

Aggregate probability	Class label	Rank
0.27125	0	3
0.26725	0	2
0.30725	0	5
0.30325	1	4
0.113044	0	1
0.32275	1	6

AUC score between class labels 0 (negative) and 1 (positive) is = AUC (0, 1)

$$A(0,1) = \frac{S - \frac{n_0(n_0+1)}{2}}{n_0 * n_1}$$

$$A(0,1) = \frac{(4+6) - \frac{(2*3)}{2}}{2*4}$$

A(0,1) = (10-3)/8 = 7/8 = 0.875

TABLE-8 Proposed technique based AUC computations for 0 and 2 classes

Aggregate probability	Class label	Rank
0.27125	0	2
0.26725	0	1
0.30725	0	3
0.36275	2	5
0.35875	2	4

$$A(0,2) = \frac{(5+4) - \frac{(2*3)}{2}}{2*3}$$

A(0, 2) = (9 – 3)/6 = 6/6 = 1

TABLE-9 Proposed technique based AUC computations for 1 and 2 classes

Aggregate probability	Class label	Rank
0.30325	2	2
0.113044	1	1
0.32275	2	3
0.36275	2	5
0.35875	2	4

$$A(1,2) = \frac{(2+3+5+4) - \frac{(4*5)}{2}}{4*1}$$

A(1, 2) = (14 – 10)/4 = 4/4 = 1

AUC for multi class is $M = \frac{2}{c(c-1)} \sum_{i<j} A(i,j)$

$$M = \frac{2}{3(3-1)} * [A(0,1) + A(0,2) + A(1,2)]$$

$$M = \frac{2}{3*2} * (0.875 + 1 + 1)$$

M = 2.875/3 = 0.958333

VII. CONCLUSION

In this paper a new technique is proposed for improving the performance measure of the probability estimation of decision trees. Proposed technique is compared with the existing best technique and the experimental results have shown that the new technique is far better than the existing technique. Experiments are conducted only for a single decision tree model but not for bagging or forest model. In the future experiments will be conducted for forests or bagging of decision trees.

REFERENCES

1. Alina Beygelzimer, John Langford, and Yuri Lifshits, “Conditional Probability Tree Estimation Analysis and Algorithms”, UAI 2009
2. Andrew J. Sage, Ulrike Genschel, Dan Nettleton, “Tree aggregation for random forest class probability estimation”, First published:02 January,2020,https://doi.org/10.1002/sam.11446,Funding information Howard Hughes Medical Institute.
3. Cesar Ferri, Peter A. Flach, and Jose Hernandez-Orallo, Improving the AUC of Probabilistic Estimation Trees”,
4. C. Ferri, P. Flanch, and J. Hernandez-Orallo, “Decision Trees for Ranking: Effect of new smoothing methods, new splitting criteria and simple pruning methods”, Department of Sistemes Informàtics i Computació, Univ. Politècnica de València, Spain, and Department of Computer Science, University of Bristol, UK
5. Charless X. Ling and Robert J. Yan, “Decision Tree with Better Ranking”, Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003) Washington DC 2003
6. Cheng Zhang and Frederick A. Matsen, “Generalizing Tree Probability Estimation via Bayesian Networks”, 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.
7. Dragos and D. Margineantu, “Improved Class Probability Estimates from Decision Tree Models”, Department of Computer Science, Oregon State University
8. Foster Provost and Pedro Domingos, “Tree induction for probability based ranking”, Machine Learning, 52, 199–215, 2003, Kluwer Academic Publishers. Manufactured in The Netherlands.
9. HENRIK BOSTRÖM, “FORESTS OF PROBABILITY ESTIMATION TREES”, International Journal of Pattern Recognition and Artificial IntelligenceVol. 26, No. 02, 1251001 (2012)Regular Papers: Machine Learning.
10. HAN LIANG, HARRY ZHANG, YUHONG YAN, “DECISION TREES FOR PROBABILITY ESTIMATION: AN EMPIRICAL STUDY”, **PUBLISHED IN:** 2006 18TH IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE (ICTAI'06)
11. Isabelle Alvarez , Stephan Bernard, and Guillaume Deffuant, “Keep the Decision Tree and Estimate the Class Probabilities Using its Decision Boundary”, IJCAI-07
12. Khan Z and Gul A, Mahmoud O, Miftahuddin M, Perperoglou A, Adler W, and Lausen, “An Ensemble of Optimal Trees for Class Membership probability estimation”, (2016), Springer International Publishing Switzerland 2016.
13. Kun Zhang and Bill P. Buckless, “Probability estimation trees: empirical comparison, algorithm extension and applications”, Publisher Tulane University, Computer Science Dept. School of Engineering New Orleans, LA, United States
14. Nitesh V. Chawla and David A. Cieslak, “Evaluating Probability Estimates from Decision Trees”, Article · January 2006, Research Gate, Copyright 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.
15. Peter Flach and Edson Takashi Matsubara, “On classification, ranking, and probability estimation”, Dagstuhl Seminar Proceedings 07161, Probabilistic, Logical and Relational Learning - A Further Synthesis, http://drops.dagstuhl.de/opus/volltexte/2008/1382
16. Rivka Levitan and Haimonti Dutta, “Improving Probability Estimation Trees for Ranking”, Center for Computational Learning Systems at Columbia University, New York

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.