Chapter 3

Performance analysis of Decision Tree Classifier before Pruning and after Pruning
CHAPTER 3

PERFORMANCE ANALYSIS OF DECISION TREE CLASSIFIER BEFORE PRUNING AND AFTER PRUNING

3.1 Introduction

Decision trees are data mining technology that has been around in a form very similar to the technology of today for almost twenty years, and early versions of the algorithms date back to the 1970's [97-99]. Often, these techniques were originally developed for statisticians to automate the process of determining which fields in their database were actually useful or correlated with the particular problem that they were trying to understand. Partially because of this history, decision tree algorithms tend to automate the entire process of hypothesis generation and then validation much more completely and in a much more integrated way than any other data mining techniques. They also particularly adept at handling raw data with little or no preprocessing. Perhaps also because they were originally developed to mimic the ways an analyst interactively performs data mining, they provide a simple-to-understand predictive model based on rules.

Because decision tree score so highly on many of the critical features of data mining, they can be used in a wide variety of business problems for both exploration and prediction. They have been used for problems ranging from credit card attrition prediction to time series prediction of the exchange rate of different international currencies [100]. Usually, the models to be built and the interactions to be detected are much more complex in real-world problems, and this is where decision trees excel.

This chapter discusses the performance of Decision tree classifier before pruning and after pruning based on accuracy. It also compares the accuracy of Decision tree classifier built using the random samples which satisfy the threshold with the one which doesn't satisfy the threshold. Based on the experimental result, a best performance Decision tree classifier has been identified and it is allowed to participate in DCMCS.
Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf node represent classes or class distributions. The top most node in the tree is the root node. Leaf nodes give a classification, that applies to all instances that reach the leaf, or a set of classification, or a probability distribution over all possible classification. To classify an unknown instance, it is routed down the tree according to the values of the attributes tested in successive nodes, and when a leaf is reached the instance is classified according to the class assigned to the leaf.

In decision tree construction process, the important thing which is to be considered is the selection of attribute used for splitting the example set in to different classes. The attribute is selected at each non-leaf node based on the information and gain. The attribute which produced the highest information is selected for splitting. Associated with node of a tree, information represents the expected amount of information that would be needed to specify whether a new instance should be classified in different classes [101].

The detailed discussion of information and gain is in the following section. The different class label’s are shown at the leaves. A typical general decision tree is shown in figure 3.1. Any node that contains class label will not have to be split further, and the recursive process down that branch will terminate. In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node that holds the class prediction for that sample. Decision trees can easily be converted to classification rules.
3.2 Needs of decision trees

Decision tree classification techniques have been used for a wide range of classification problems and becoming an increasingly important tool for classification. The needs of decision tree for today’s applications are as follows.

- To reliably apply the segmentation scheme to a set of data that reflects a group of potential customers.
- To identify possible interactive relationship between variables in a way that would lead to understand how changing one variable can affect another.
- To provide a visual representation of the relationships between variables in a way that would lead you to understand how changing one variable can affect another.
- To provide a visual representation of the relationship between variables in the form of a tree, which is relatively an easy way to understand the nature of the data residing in the databases.
- To explore data to identify important variables in a data set that can eventually be used as a target [97].
3.3 Overfitting

When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. In result, poor accuracies have been obtained for unseen data. Tree pruning methods address this problem of overfitting the data. Such methods typically use statistical measures to remove the least reliable branches, generally resulting in faster classification and an improvement in the ability of the tree to correctly classify independent test data. The designer may not be aware of the existence of outliers at the time of the tree design, even though outliers are usually easily detected in a training set; in some cases a thorough examination of data set may be necessary.

3.4 Tree pruning

There are two common approaches to the pruning. They are pre-pruning and post-pruning. In the pre-pruning approach, a tree is “pruned” by halting its construction early (e.g., by deciding not to further split or partition the subset of training samples at a given node). Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset samples or the probability distribution of those samples.

The second approach, post-pruning, removes branches from a “fully grown” tree. A tree node is pruned by removing its branches. The lowest unpruned node becomes a leaf and is labeled by the most frequent class among its former branches. For each non-leaf node in the tree, the algorithm calculates the expected error rate that would occur if the sub tree at that node were pruned. Next, the expected error rate occurring if the node were not pruned is calculated using the error rates for each branch, combined by weighting according to the proportion of observations along each branch. If pruning the node leads to a greater expected error-rate, then the sub tree is kept. Otherwise, it is pruned. After generating a set of progressively pruned trees, an independent test set is used to estimate the accuracy of each tree. The decision tree that minimizes the expected error rate is preferred. Post-pruning requires more computation than pre-pruning yet generally leads to a more reliable tree.
3.5 Splitting Indices

3.5.1 Entropy

Entropy provides an information-theoretic approach to measure the goodness of a split. Assume that there are \( n \) equally probable possible messages. The probability \( p \) of each message is \( 1/n \) and hence, the information conveyed by a message is \(-\log_2(p) = \log_2(n)\). If there are 4 messages then since \( \log(16) = 4 \), we need 4 bits to identify each message [104-106].

If we are given a probability distribution \( P = (p_1, p_2, \ldots, p_n) \), then the information conveyed by this distribution, also called the entropy of \( P \), which is given by

\[
\text{Entropy (P)} = - [ p_1 \log(p_1) + p_2 \log(p_2) + \ldots + p_n \log(p_n) ]
\] (3.1)

In the context of decision trees, if the outcome of a node is to classify the records into two classes, \( C_1 \) and \( C_2 \), the outcome can be viewed as a message that is being generated and the entropy gives the measure of information for a message to be \( C_1 \) or \( C_2 \). If a set of records \( 'T' \) is partitioned into a set of disjoint exhaustive classes \( C_1, C_2, \ldots, C_n \) on the basis of the value of the class attribute, then the information needed to identify the class of an element of \( 'T' \) is

\[
\text{Info (T)} = \text{Entropy (P)}
\] (3.2)

where \( P \) is the probability distribution of the partition \( C_1, C_2, \ldots, C_n \) and \( P \) is computed based on their relative frequencies, i.e.,

\[
P = \left( \begin{array}{ccc}
|C_1| & |C_2| & |C_n| \\
\frac{1}{|T|} & \frac{1}{|T|} & \ldots \frac{1}{|T|}
\end{array} \right)
\] (3.3)
3.5.2 Information for a partition

If T is partitioned, based on the value of the non-class attribute X, into sets \( T_1, T_2, \ldots, T_n \), then the information needed to identify the class of an element of T becomes the weighted average of the information to identify the class of an element of \( T_i \) [107]. The weighted average of \( \text{info}(T_i) \) is given as

\[
\text{Info}(X, T) = \frac{1}{|T|} \sum_{i=1}^{N} \frac{|T_i|}{|T|} \text{info}(T_i)
\]

(3.4)

3.5.3 Gain

Gain is defined as the differences between the entropy of the original segment and accumulated entropies of the resulting split segment [107]. The information gain due to a split on attribute X is defined as

\[
\text{Gain}(X, T) = \text{info}(T) - \text{info}(X, T).
\]

(3.5)

and the gain-ratio(X,T) is given by

\[
\text{Gain-ratio}(X, T) = \frac{\text{Gain}(X, T)}{\text{Info}(X, T)}
\]

(3.6)

The information gain represents the differences between the information needed to identify an element of T after the value of attribute X is obtained. That is, the information gains due to an attribute X. The above concepts are used in the decision tree construction algorithm.
3.6 Algorithm
To generate a decision tree from the given training data the Enterprise Data-Miner (EDM) is used. The EDM data mining tool is explained in appendix 1.

**Input:** Training samples, the set of candidate attributes and attribute-list.

**Output:** a decision tree

**Method**
1) create a node N;
2) if samples are all of the same class, C then
3) return N as a leaf node labeled with the class C;
4) if attribute-list is empty then
5) return N as a leaf node labeled with the most common class in samples;
6) select test-attribute, the attribute among attribute-list with the highest information gain;
7) label node N with test-attribute;
8) for each known value a_i of test-attribute
9) grow a branch from the node N for the condition test-attribute = a_i
10) let s_i be the set of samples in samples for which test-attribute = a_i
11) if s_i is empty then
12) attach a leaf labeled with the most common class in samples;
13) else attach the node returned by generate_decision_tree(s_i,attribute-list-test-attribute);

3.7 The Best attribute for the Classification
In the Decision tree construction process, the best attribute for the classification has been chosen as follows
1) Each non leaf node of a decision tree corresponds to an input attribute, and each are to a possible value of that attribute. A leaf node corresponds to the expected
value of the output attribute when the input attributes are described by the path from the root node to that leaf node.

2) In a good decision tree, each non leaf node should correspond to the input attribute which is most informative about the output attribute amongst all the input attributes not yet considered in the path from the root node to that node. This is because we would like to predict the output attribute using the smallest possible number of questions on average.

3) Entropy is used to determine how informative a particular input attribute is about the output attribute for a subset of the training data. Entropy is a measure of uncertainty in communication systems. It is a fundamental in modern information theory [108].

3.8 Stopping Criteria

A decision tree construction process is concerned with identifying the splitting attributes and splitting criteria at every level of the tree. If samples at a node belong to two or more classes, then a test is made at the node that will result in a split. This process is recursively repeated for each of the new intermediate nodes. The recursive process stops only when any one of the following condition is true.

a) All samples for a given node belong to the same class.

b) There are no remaining attributes on which the samples may be further partitioned. In this case, majority voting is employed. This involves converting the given node into leaf and labeling it with the class in majority among samples. Alternatively, the class distribution of the node samples may be stored.

c) There are no samples for the branch. In this case, a leaf is created with the majority class in samples [108].

3.9. Potentials and Problems with Decision Tree Classifier

Decision tree classifiers (DTC's) are attractive for the following reasons
1) Global complex decision regions (especially in high-dimensional spaces) can be approximated by the union of simpler local decision regions at various levels of the tree.

2) In conventional single-stage classifiers where data sample is tested against all classes, which reduces the efficiency. Where as in a tree classifier a sample is tested against only certain subsets of classes, thus eliminating unnecessary computations.

3) In single stage classifiers, only one subset of features is used for discriminating among all classes. This feature subset is usually selected by a globally optimal criterion, such as maximum average interclass separability. On the other hand, in decision tree classifiers one has the flexibility of choosing different subsets of features at different internal nodes of the tree such that the feature subset chosen optimally discriminates among the classes in that node. This flexibility may actually provide performance improvement over a single-stage classifier.

4) In multivariate analysis, with large numbers of features and classes, one usually needs to estimate either high-dimensional distributions (possibly multi-modal) or certain parameters of class distributions, such as a priori probabilities, from a given small sized training data set. In so doing, one usually faces the problem of "high-dimensionality." This problem may be avoided in a DTC by using a smaller number of features at each internal node without excessive degradation in the performance.

5) The classification process can be done in $O(\log_2(N))$ instead of $O(N)$ where $N$ is the number of classes

6) A tree classifier is more flexible than a single-stage one in that the nodes can have different decision rules and different features. In single-stage classifiers, a
subset of features and the decision rule are selected by optimizing a global criterion and are used to discriminate among all classes. In contrast, a tree classifier offers the flexibility to select different features and a unique decision rule for each node [109-114].

The possible drawbacks of DTC, on the other hand, are:

1) Overlap especially when the number of classes is large, can cause the number of terminals to be much larger than the number of actual classes and thus increase the search time and memory space requirements. Errors may accumulate from level to level in a large tree. It is pointed out by Wu et.al. [113] that one cannot simultaneously optimize both the accuracy and the efficiency; for any given accuracy a bound on efficiency must be satisfied.

2) Finally, there may be difficulties involved in designing an optimal DTC. The performance of a DTC strongly depends on how well the tree is designed.

3) A tree classifier may circumvent the effects due to small training sample size by focusing on fewer classes and using fewer features at each node. In particular, there are fewer nodes near the root of the tree and consequently more training samples are available per node. The tradeoff for these benefits is a more complex design process.

4) Most of the algorithms (like ID3 and C4.5) require that the target attribute will have only discrete values. As decision trees use "divide and conquer" method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present. One of the reasons for that is that other classifiers can compactly describe a classifier that would be very challenging to represent using a decision tree. A simple illustration of this phenomenon is the replication problem of decision trees [109]. Since most decision trees divide the instance space into mutually exclusive regions to represent a concept, in some cases
the tree should contain several duplications of the same sub tree in order to represent the classifier. The greedy characteristic of decision trees leads to another disadvantage that should be pointed out that is its over-sensitivity to the training set, to irrelevant attributes and to noise [114].

3.10 Experimental Results

3.10.1 Adult Database

The statistical detail of adult database is discussed in chapter 2. In Adult database, random sampling has been performed in 10 different iterations. The random samples obtained in the 10th iteration only satisfies the threshold for all the classes. In decision tree construction process, the random samples obtained in all iterations have been used and individual class accuracies were estimated. Table 3.1 shows the Accuracy of Decision tree classifier for the individual classes.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Accuracy before Pruning</th>
<th>Accuracy after Pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class 1</td>
<td>Class 2</td>
</tr>
<tr>
<td>1</td>
<td>60.13</td>
<td>74.12</td>
</tr>
<tr>
<td>2</td>
<td>60.91</td>
<td>74.34</td>
</tr>
<tr>
<td>3</td>
<td>61.56</td>
<td>74.01</td>
</tr>
<tr>
<td>4</td>
<td>60.78</td>
<td>74.34</td>
</tr>
<tr>
<td>5</td>
<td>60.07</td>
<td>74.15</td>
</tr>
<tr>
<td>6</td>
<td>59.94</td>
<td>74.63</td>
</tr>
<tr>
<td>7</td>
<td>60.23</td>
<td>73.96</td>
</tr>
<tr>
<td>8</td>
<td>61.12</td>
<td>74.03</td>
</tr>
<tr>
<td>9</td>
<td>61.04</td>
<td>74.17</td>
</tr>
<tr>
<td>10</td>
<td>62.07</td>
<td>75.52</td>
</tr>
</tbody>
</table>
The experiments have been carried out for the 10 different random samples in the adult database. Initially, the trees were constructed and accuracies were estimated for the different random samples. After that, post-pruning has been performed on the fully grown tree and accuracies were estimated after pruning for the same set of trees. Figure 3.2 compares the overall accuracy of decision tree classifier for the Adult database before pruning and after pruning in different iterations. The Figure shows that 3 to 6 percent improvement in accuracy after pruning. The class 1 for the Adult database is “>50K” and class 2 for the adult database is “<=50K”. Class 1 represents cases that those peoples annual income is above 50K and class 2 are the cases that those peoples annual income is less than or equal to 50K.

Figure 3.2 Comparing the Accuracy of Decision tree Classifier before Pruning and after Pruning using Adult database

Figure 3.3 compares the accuracies of these two classes in different iterations after pruning. It also shows that the accuracy of class 2 is always outperforms the accuracy of class 1. The accuracy of class 1 is in the range 63% to 66% and the accuracy of class 2 is in the range 78% to 81%. The accuracy of class 1 is less than class 2 in all 10 iterations. The reason is that the number of samples in class 1 is less by 2000 to 5000 samples than in class2.

Figure 3.3 also shows that a class which doesn’t satisfy the threshold in an iteration have produced less accuracy than the same class which satisfy the threshold
in another iteration. From the 10 iterations, the 10th iteration only satisfies the threshold for all the classes. The individual class accuracies of all the classes have been improved in the 10th iteration only.

![Accuracy Graph](image)

**Figure 3.3 Individual Class performance of Decision Tree after Pruning for the Adult database**

Figure 3.4 shows the ROC Curve for the Decision tree classifier. It shows the relationship between false positives and true positives. The curve was originally used to examine the percentage of retrieved that are not relevant versus the percentage of retrieved that are relevant. A ROC graph is a plot with the false positive rate on the X axis and the true positive rate on the Y axis.

![ROC Curve](image)

**Figure 3.4 ROC Curve of Decision tree Classifier for the Adult database after Pruning**
The point \((0,1)\) is the perfect classifier: it classifies all positive cases and negative cases correctly. The point \((1,1)\) corresponds to a classifier that predicts every case to be positive. The point \((0,0)\) represents a classifier that predicts all cases to be negative, while Point \((1,0)\) is the classifier that is incorrect for all classifications. The False positive range obtained for the Decision tree classifier for the adult database has been varying from 0.16 to 0.2 in different iterations. The true positive range has been obtained in the range from 0.80 to 0.84 for the different random samples. In all iterations the X-Y plot of false positive and true positive range is with the accepted region only. Even though the individual class accuracies doesn’t satisfy the threshold in the first nine iterations, the performance of Decision tree classifier in ROC graph is near the perfect classifier.

3.10.2 Earthquake database

The statistical details of Earthquake database is discussed in chapter 2. The earthquake database consists of three classes such as class 1, class 2 and class 3. The cases of those magnitude from 4.0 to less than 5.5 is in class 1, the magnitude from 5.5 to less than 7 is in class 2 and the magnitude from 7 and above is in category 3.

The threshold have been computed and fixed for the training data set as shown in Table 2.7. The random sampling has been performed on the Earthquake database in 12 different iterations. Table 3.2 shows the Accuracy of Decision tree classifier for the individual classes. The experiments have been carried out for using the 12 different random samples for the adult database. The post pruning have been performed on the fully grown tree and the accuracies were computed before pruning and after pruning. In every iteration, the accuracies have been estimated for the individual classes before pruning and after pruning.

Figure 3.5 compares the overall accuracy of decision tree classifier before pruning and after pruning for the Earthquake database. The Figure shows that there is a 4 to 6 percentage of improvement in accuracies in the individual classes and also in the overall accuracy after pruning.
Table 3.2: Individual Class performance of Decision Tree before Pruning and after Pruning for the Earth Quake database

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Accuracy before Pruning</th>
<th>Accuracy after Pruning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Class 1</td>
<td>Class 2</td>
</tr>
<tr>
<td>1</td>
<td>82.65</td>
<td>60.18</td>
</tr>
<tr>
<td>2</td>
<td>83.12</td>
<td>64.56</td>
</tr>
<tr>
<td>3</td>
<td>82.14</td>
<td>64.34</td>
</tr>
<tr>
<td>4</td>
<td>82.45</td>
<td>62.12</td>
</tr>
<tr>
<td>5</td>
<td>83.65</td>
<td>64.16</td>
</tr>
<tr>
<td>6</td>
<td>82.75</td>
<td>62.59</td>
</tr>
<tr>
<td>7</td>
<td>82.92</td>
<td>63.14</td>
</tr>
<tr>
<td>8</td>
<td>82.45</td>
<td>64.23</td>
</tr>
<tr>
<td>9</td>
<td>82.23</td>
<td>65.16</td>
</tr>
<tr>
<td>10</td>
<td>82.13</td>
<td>61.45</td>
</tr>
<tr>
<td>11</td>
<td>83.17</td>
<td>63.02</td>
</tr>
<tr>
<td>12</td>
<td>84.23</td>
<td>66.24</td>
</tr>
</tbody>
</table>

Figure 3.5: Comparing the Accuracy of Decision tree Classifier before Pruning and after Pruning using Earthquake database
Figure 3.6 compares the accuracies of different classes in different iterations for the Earthquake database after pruning. It also shows that the accuracy of class 1 always outperforms the accuracy of class 2 and class 3 and the accuracy of class 2 outperforms class 3 in most of the iterations. The accuracy of class 1 is in the range 86% to 89%, the accuracy of class 2 is in the range 65% to 75% and that of class 3 is in the range 56% to 73%. The reason is that the number of samples in class 1 is more than class 2 and class 3 and also the number of classes in class 2 is more than class 3. So the percentage of accuracy is less for class 3 and more for class 1.

Figure 3.6 also shows that classes that didn't satisfy the threshold in an iteration have produced less accuracy than the one which satisfy the threshold in another iteration. From the 12 iterations, the 12th iteration only satisfies the threshold for all the classes. The individual class accuracies of all the classes have been improved in the 12th iteration only. Figure 3.7 shows the ROC Curve for the Decision tree classifier using the Earthquake database. It shows the relationship between false positives and true positives.
The False positive range obtained for the Decision tree classifier for the earthquake database has been varying from 0.08 to 0.125 in different iterations. The true positive range has been obtained in the range from 0.875 to 0.925 for the different random samples. In all iterations the X-Y plot of false positive and true positive range is with the decision region only. Even though the individual class accuracies does not satisfy the threshold in the first 11 iterations, the performance of Decision tree classifier in ROC graph is near the perfect classifier.

3.11 Conclusion

Top down induction of decision trees is probably the most extensively studied method of machine learning used in data mining. Decision trees are self-explanatory and when compacted they are also easy to follow. Furthermore decision trees can be converted to a set of rules. Thus, this representation is considered as comprehensible[97].

Two datasets have been used for the experiments. In the Adult data set, the random samples have been selected for 10 times and experiments have been carried out on each random sample. The accuracies have been computed before pruning and after pruning. The results show that in all iterations, the accuracy of Decision tree after pruning is always increased than the accuracies before pruning for both the database.
The individual Class performances of Decision Tree after Pruning for both databases clearly show that the accuracies of individual classes are less if the number of samples in that class is less for that iteration. Moreover, the accuracies of all the classes in the last iteration are more than the accuracy of individual classes in the previous iterations. This is because of the last iteration only satisfies the threshold for all the classes.

The ROC curves shown in Figures 3.4 and 3.7 shows the performance of Decision tree classifier for the Adult database and Earthquake database respectively. Both figures show that the performance of decision tree classifier is satisfied for all iterations. But the ROC curve does not show the individual class performance. Even though the ROC curve shows the good performance for Decision tree classifier, the individual class performance is not satisfactory for both the databases.

Figure 3.8 compares the performance of Adult database and the earthquake database. The figure shows that the Accuracy obtained by Adult database is higher than the earthquake database in different iterations. In the adult database, 33% of the attributes are continuous. But in the earthquake database, 88% of the attributes are continuous. Hence the Decision tree have produced more accuracies to the database containing categorical attributes than the database containing continuous attributes.

When considering the quality of input for decision trees, it is good in handling number of records. The ability to handle large number of attributes for decision tree is also good. They are capable of handling both nominal and numeric input attributes. It has poor performance in handling strings. Decision tree representation is rich enough to represent any discrete-value classifier. They are capable of handling datasets that may have errors and missing values. Decision trees are considered to be a nonparametric method. This means that decision trees have no assumptions about the space distribution and on the classifier structure.
Decision trees have the ability to learn transparent rules and the ability to estimate statistical significance. It has poor performance in ability to learn incrementally. The application performance of the decision tree is good and the learning performance is average. The computing time is less for the Decision tree classifier than the maximum likelihood classifier and by comparison, the statistical errors are avoided.

In learning process, different criteria for attribute selection, or modified pruning methods are used. In our discussion of decision trees, we have used only one attribute for splitting the data into subsets at each node of the tree. However, it is possible to allow test that involves several attributes at a time.

Perhaps one of the main features of DTC's is the flexibility they provide; for example the capability of using different feature subsets and decision rules at different stages of classification and the capability of trade-offs between classification accuracy and time/space efficiency.
Based on the experiments, a best performance classifier has been identified in both the database. For the Adult database, the classifier which is constructed in the 10th iteration after pruning produces highest accuracy than the any other classifier. For the Earthquake database, the classifier constructed in the 12th iteration after pruning provides highest accuracy. These selected classifiers are allowed to construct the proposed DCMCS for the better performance.

In the next chapter, the same set of samples in different iterations have been used and the accuracies are estimated using K-Nearest neighbor classifier by varying the K size form 1 to 25.