Improved Post Pruning of Decision Trees
Roopa C1 A. Thamaraiselvi2 S. Preethi Lakshmi3 Mr. N. Bhaskar4
1,2,3,4Department of Information Technology
1,2,3,4KCG College of Technology, Chennai-97, India

Abstract— Decision trees are strong predictors which are used to explicitly represent large data sets. An efficient pruning method will prune or eliminate the non-predictive parts of the model and generate a small and accurate model. This paper presents an overview of the issues present in decision trees and the pruning techniques. We evaluated the results of our pruning method on a variety of machine learning data sets from UCI machine learning repository and found that it generates a concise and accurate model.

Key words: Clustering, Decision Tree, Pruning, Data Mining

I. INTRODUCTION

Decision trees are predicative tools useful in data mining applications. Decision trees are used to explicitly represent data sets in a tree structure, each node denotes a test on an attribute and each branch represents an outcome of that test. The topmost node in a tree is the root node. Decision tree classifiers are popular because they have good accuracy, construction of a decision tree doesn’t require any specific domain and decision trees can be quickly generated for high dimensional data. The generated decision trees should predict accurately and have the ability to analyze the model. It is important that the constructed tree be accurate and easy to interpret.

When a decision tree is built many of the branches will reflect anomalies due to noise or outliers. Ideally, tree pruning should eliminate the non-predictive parts of the model but never eliminate any part that is truly predictive. Tree pruning method addresses the problem of overfitting the data. It uses statistical measures to remove the least reliable branch making the tree smaller and less complex.

Statistical significance tests are used to make sure that the achieved results are statistically valid and not a product of comparison of procedures. Two important measures are checked, first is the probability that the relationship exists; and second is if it exists, how strong is the relationship. There are too many sources of error, for example, sampling error, researcher bias, problems with reliability and validity and even simple mistakes. Statistical significance test is not the same as practical significance test.

There are two common approaches to tree pruning namely: pre- and post- pruning. The pre-pruning approach is implemented in the growing phase, before it perfectly classifies the training set by halting its construction early. Halting that node becomes a leaf node. An additional stopping criterion is built for this purpose. Post pruning removes sub trees from a fully grown tree, allowing the tree to perfectly classify the training set and then post prune the tree. A sub tree at a given node is pruned by removing its branches and replacing it with a leaf node.

The remainder of this paper is organized as follows. Section 2 introduces basics of decision trees. Section 3 compares the pruning techniques. Section 4 states the proposed model. Section 5 Evaluation. Section 6 Conclusion and future work. Section 7

II. BASIC CONCEPTS

Tree induction algorithm take some data collected from a domain to build a decision tree model. The individual data observation is called data instance and the entire set of observations is called data set. Each observation is characterized by an attribute. The attributes can be of any type; numeric or alphabet.

A. Classification and Prediction

In data mining applications, very large training sets of million tuple are common. Decision trees can be used regression or classification. In Traditional decision trees each node represents a single classification. In classification each node represents a cluster or a sub tree. The data classification process involves building the model and using the classifier for classification. A classifier is constructed to predict categorical labels, such “yes” or “no”; “high” “medium” or “low”, or other class labels. The classifier is built from training set taken from the database tuples and their class labels.

In prediction, the model constructed predicts a continuous-valued function, or a value as opposed to a categorical label. The major issue in classification and prediction is in preparing the data.

B. Decision Trees:

Decision trees are constructed in a top-down recursive manner. Decision tree induction is the learning of decision trees from class labeled training tuples or training sets from a

<table>
<thead>
<tr>
<th>Income</th>
<th>Business</th>
<th>Student</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>High</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Low</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Low</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

C. An Example Data Set:

Fig. 1: Example of a Decision Tree
D. A Decision Tree:

Large example data set. The training set is recursively partitioned into smaller subsets based on the classifier as the tree is being built. The recursive partitioning stops until all of the training sets are covered or there are no tuples for a given branch. Every internal node is labeled with the name of a predicate attribute in case of a non-leaf; the branches coming out from an internal node are given one of the values of the attribute of the labeled node and a leaf node is labeled with a class.

Decision tree induction is related to rule induction where each path from the root node of the generated decision tree to one of its leaf node can be transformed into a rule by conjunction the tests along that path. From, figure 1. For example a path can be transformed, “If income is high, i.e. greater than 45,000 then individual is likely to be in business”.

Decision tree induction technique helps to generate small trees that fit a training set of data. One of the factors that determine the success of pruning a decision tree is its optimal size and providing the largest information gain to the user. A full analysis of a decision tree comprises of two steps, growing a tree and pruning a tree. In the following section a detailed review of the various tree pruning methods are analyzed.

III. A REVIEW OF PRUNING TECHNIQUES

An overview of the different pruning techniques and issues involved are briefly discussed. Tree pruning is used to address the problem of over fitting the data. Pruned trees tend to be small, less complex and easier to understand. There are two common approaches to pruning:

- Pre-pruning, this approach deals with noise while the tree is still growing
- Post-pruning, allows the tree to be completely classified.

The pruning methods considered here are post pruning methods, even though pre-pruning requires less computation it typically does not avoid the problem of overfitting and leads to rather unreliable trees. Another reason is that it runs the risk of missing future splits that occur under the weaker splits.

A. Post Pruning:

There are two main methods of post pruning a tree. One way is to extract rules from a decision tree, where each rule is created for each path from the root to the leaf node. Each splitting criterion along a given path is logically ANDed to form the rule antecedent i.e. "IF part" and the consequent forming the "THEN part". Another method is to retain the decision tree but to eliminate the non-predictive parts and replace it with a leaf node. An optimal tree is formed by minimizing the generalization errors, some of the post pruning techniques are: error-based pruning [1], reduced-error [1], cost-complexity pruning [1] and pessimistic pruning [1].

1) Error-Based Pruning:

Error-based (ERB) pruning is a simple method that does not require a validation set. It is implemented in well-known C4.5 algorithm. Consider some E errors at N training examples at a leaf node of the tree to give an estimate of the error probability of that node. Using the binomial distribution theorem, the confidence limits can be calculated for the probability of error of a given confidence level.

\[
CDF = \sum_{k=0}^{N} \binom{N}{k} p^k q^{N-k}
\]  

From this we can observe the probability of error for the entire example training sets that will end up at a leaf node. The number of predicted errors of a subtree is the sum of the predicted errors of its branches, and if the number of predicted errors for a leaf is less than the number of predicted errors of a subtree of that leaf then that subtree is replaced with that leaf. From a study on empirical study on decision tree pruning, it was reported that ERB under-prunes on all the data set that they tested [1].

2) Reduced-Error Pruning:

It is a simple procedure for pruning decision trees, suggested by Quinlan (1987) [1]. Reduced-Error pruning (REP) requires a separate pruning set. It is one of the simplest forms of pruning where a non-leaf node is considered a contender for pruning, where each internal node can be replaced with a leaf node based on the error rate over the pruning set. If the error rate of the new tree is equal to or smaller than the original tree and that subtree contains no subtrees with that property, then that subtree is replaced with a leaf node, else do not prune. One of the main advantage is its simplicity and speed. But when the test set is much smaller than the training set [1], it leads to over pruning.

3) Cost Complexity Pruning:

This approach is a CART (Classification and Regression Trees) standard, that assigns a unit of cost for misclassification of instances, this misclassification cost is used to reflect the seriousness of errors due to misclassification. The cost complexity of a tree T is a function of the number of leaves in the tree and the error rate of the tree. It is a bottom up approach where for each non-leaf node N, the cost complexity of the subtree at that node is calculated. The sum of all the costs is the equivalent cost complexity of the decision tree. This algorithm generates a sequence of pruned trees and the smallest tree that minimizes the cost complexity is preferred.

- Take the fully grown decision tree for any two node which descend from the same node, if the error rate is the same or greater than the parent prune off the two nodes.
- This process is followed recursively and the tree shrinks little by little.
- We continue the same thing until we cannot find any nodes which do not satisfy this equality.

Though misclassification cost is calculated, it is difficult to choose the cost for different errors [1]. The final tree generated can be largely dependent on that subjective choice.

4) Pessimistic Pruning:

It is very similar to cost complexity pruning, it also uses error rate estimates, to make decisions for subtree pruning. A prune set or cross validation set is not required, it uses the training set to estimate the error rate. If an internal node is pruned, then all its descendents are pruned, leading to fast pruning.
IV. PROPOSED WORK

In this paper we propose an improved method for post pruning of decision trees. The main aim is to avoid overfitting and over pruning the tree. There are two phases: first is to generate a tree from the example training set and the second is to apply post pruning method to eliminate the non-predictive parts of the decision tree.

A. Data Cleaning:

Data cleaning can be considered the first step of any data analysis, to ensure the quality of the data. Data cleaning means to "clean" the data by filling in the missing values, smoothing the noisy data, identifying or removing the outliers and to resolve inconsistencies. It also includes data integration, data transformation and data reduction. Data integration is the process of integrating multiple databases or files and to remove redundant data across the collected data. Data transformation is an additional pre-processing procedure, to improve the success of mining. Normalization and Aggregation are such kind of operations. Data reduction technique is used to represent the data sets in a reduced volume, yet produces the same or almost same analytical result, some of the techniques are data aggregation and attribute selection.

B. Attribute Selection:

Attribute selection measure, is used for selecting the splitting criteria that best separates a given data set. It is a process of removing the redundant attributes that are irrelevant to the data mining task [1]. Redundant are those which provide no more information than the currently selected attributes and irrelevant attributes provide no useful information in that context. Several measures are available for attribute selection.

1) Information Gain:

It is the information gained on splitting a node, during the tree generation process. Information gain be positive or zero, but never negative. The reduction in uncertainty when choosing the first branch. It is the expected reduction in entropy caused by portioning the instances according to a given attribute. Where D is the data set.

\[
\text{Info}(D) = -\sum_{i=1}^{m} p(i) \log_2 p(i) \tag{2}
\]

2) Gain Ratio:

The information gain measure is biased towards tests with multiple outcomes. Gain ratio overcomes this bias, by applying normalization to information gain using split information.

\[
\text{Gain} (A) = \text{Info} (D) - \text{Info}_A(D) \tag{3}
\]

\[
\text{Gain ratio (A)} = \text{Gain} (A) / \text{Split Info} (A) \tag{4}
\]

3) Gini Index:

The gini index measures the impurity of a data set D or training data. It determines how often a randomly chosen element from a data set would be incorrectly labeled, if it were labeled according to the distribution of labels in a subset.

\[
\text{Gini} (D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{5}
\]

C. Building and Pruning a Tree:

We use a One-Class clustering method which divides the training set into a growing set and pruning set. The growing set is used to generate the tree as well as prune it, while the pruning set is used to select the best tree.

1) Algorithm:

To generate decision tree from the training data sets.

1) Step-1: Consider the records in Data set D, create a node N if tuples in D are all of the same class C.
2) Step-2: Return N as the leaf node with the class label C.
3) Step-3: If attribute list is empty then return N as the leaf node labeled with the majority class in data set D.
4) Step-4: If N contains tuples that belong to more than one class, an attribute test condition is selected to partition the record to smaller subsets.
5) Step-5: For each outcome j a child node is created and the records in N are distributed to the children based on the outcome.
6) Step-6: The algorithm is then recursively applied to each child node.

We have used the RapidMiner 5.3 tool to simulate the generation of the decision tree and implemented the pruning rules to the fully grown tree. We use C4.5-rules classification model as a method for pruning the decision tree.

1) Step-1: Check if algorithm satisfies termination criteria.
2) Step-2: Computer information-theoretic criteria for all attributes.
3) Step-3: Choose best attribute according to the information-theoretic criteria.
4) Step-4: Create a decision node based on the best attribute in step 3.
5) Step-5: Induce (i.e. split) the dataset based on newly created decision node in step 4.
6) Step-6: For all sub-dataset in step 5, call C4.5 algorithm to get a sub-tree recursively.
7) Step-7: Attach the tree obtained in step 6 to the decision node in step 4.
8) Step-8: Return the optimal tree.

V. EVALUATION
We evaluated the results of our pruning method on a variety of machine learning data sets from UCI machine learning repository and found that it generates a concise and accurate model.

Dataset description: Cause for which the user tends to use to cyber space in Kohkiloye and Boyer Ahmad Province in Iran. Collecting all the information to form a database is done by a questionnaire. The questionnaire is provided as oral, written and also programming of a website which includes a internet questionnaire and users can answer as they wish.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>left</td>
<td>political</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>political</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>middle</td>
<td>tourism</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>political</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>political</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>political</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>middle</td>
<td>news</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>medium</td>
<td>middle</td>
<td>news</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>political</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>high</td>
<td>right</td>
<td>political</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>medium</td>
<td>middle</td>
<td>tourism</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>impression</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>high</td>
<td>left</td>
<td>impression</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

Fig. 3: Shows the Data Sets of an Example Bloggers Instances

![Image](image-url)

Fig. 4: A Fully Grown Decision Tree for the Example Data Set Mr. N.Bhaskar for his continued support and to Mrs. G.Veni Devi for her insightful thoughts on our project.

VI. APPLICATION
Some of the applications we identified with our paper were with the Retail Industry, to collect data about their customers. It helps to extract the buried information from the large database.

VII. CONCLUSION AND FUTURE WORK
Several topics require future research to generate a further reduced size decision tree but still maintaining the same amount of Information Gain. To extend the implications of the paper in other practical data mining applications.

VIII. ACKNOWLEDGMENT
We would like to thank all the staffs of the Department of Information Technology, KCG College of Technology. We would like to extend our special thanks to our supervisor provided as oral, written and also programming of a website which includes a internet questionnaire and users can answer as they wish.

REFERENCES
[2] Steven W.Norton,Siemens Corporate Research,Inc.,”Generating better Decision tree”.
[3] Xinneng Zhang and Shengyi Jiang,Cisco School of Informatics,China.,”A Splitting Criteria Based on Similarity in Decision Tree Learning”, JOURNAL OF SOFTWARE, VOL.7,No.8,August-2012.


