Decision Tree Induction Strategies

Avinash S. Jagtap
Department of Statistics, Tuljaram Chaturchand College, Baramati, Dist. Pune
Maharashtra, India

Abstract

Decision trees offer a combination of classification and regression. Decision trees are grown with the purpose of reducing ambiguity and increasing certainty, where the ultimate aim is to develop a prediction model that can work beyond the available data. There are two basic approaches to grow a decision tree and these two approaches are completely opposite to each other in direction. One is the top-down approach and the second is the bottom-up approach. In addition, several options are available for branching criteria and stopping rules. This paper takes a critical review of some of the important and populartree induction methods in order to decide the most appropriate tree induction method in a specified situation. An illustrative example is used in order to bring out the conclusions, easy to understand and also to make recommendations implementable.

KEYWORDS-Decision tree, Sensitivity, Robustness, Concept Learning System (CLS).

I. Introduction

Decision tree induction is not an easy or simple task. A good decision tree must satisfy multiple requirements or possesses multiple characteristics. Further, growing an optimal decision tree is not a straight forward activity. It often requires a dynamic approach that increases computational complexity of the procedure. Nevertheless, there are two major ways to approach the problem of growing a decision tree. There are the top-down and the bottom-up approaches. The former begins at data points in one node, called root of the tree and grows the tree by branching out cases on the basis of one characteristic (or attribute) and test condition. The tree is binary if every test condition is binary. On the contrary, the latter approach begins with every observation as a leaf node and goes on joining leaf nodes that are most similar in terms of some characteristics. Moving from the bottom to the top, where all observations are put in a single node, the bottom-up decision tree is built. The attributes used in this process are of important whenever they characterize nodes at different levels of the tree. The ease of induction and interpretation havemade the top-down approach more popular and most common in practice. Nevertheless, this paper intends to review various decision tree induction methods in order to identify the most practical strategy that could balance sensitivity and robustness of the resulting decision tree.

Decision trees provide one of the most effective and, at the same time, meaningful ways of dividing data into subsets that are not only internally homogenous but also useful in dividing future data most appropriately. Decision trees appear very similar to hierarchical agglomerative cluster but simultaneously are different, as they are developed by using supervised algorithms, while the latter is developby using unsupervised algorithms. The second major difference between the two is decision trees are used for classifying yet unobserved future data, where as agglomerative hierarchical clustering is restricted to the available data and cannot be used for any future data. A decision tree is a special case of a graph. It may, therefore, be appropriate to begin by defining a decision tree from a graph theoretical point of view. Graph, theoretically speaking is a decision tree G = (V, E) consists of the finite non-empty set V of nodes (vertices) and a set E of edges. Such a graph is called tree if it satisfies the following conditions.

- Every edge e = (u, v) must be an ordered pair of vertices, making tree a directed graph.
- The graph must not contain cycles, making a tree an acyclicgraph.
- The edge e = (u, v) is an entering edge for v, while it is leaving edge for u.
- There is an unique vertex, called the root node that has no entering edge. Every other node has an exactly one entering node.
- Every node has at least two leaving edges (internal node) or no leaving edge (leaf node).
- Every node v_n has a unique path from the root and this path is defined by the sequence (v_1, v_2),(v_2, v_3) …., (v_{n-1}, v_n), where v_1 is the root.
- If there is a path from node v to node w and v ≠ w, then v is called a proper ancestor of w and w is called a proper descendent of v. A node is called a leaf node, if it has no proper descendent. All other nodes, except the root, are called internal nodes.

The root and every internal node hold a test for one of the attributes, or a set of attributes, in the given data and every leaving edge corresponds to one of the outcomes of this test. A leaf node holds a class label where the problem is of classification and an expected value (of the response variable) where the problem is of regression. In some peculiar cases, a leaf node can also hold a model, developed through machine learning algorithm.

The value of response variable can be predicted by navigating through decision tree. This is done by starting at the root and following the edges, one after another, according to the test results at successive internal nodes. Once the leaf node is reached, an informationcan be used for making desired prediction. For example, the classical decision tree has a class label for every leaf node and this is predicted class of every instance reaching the particular leaf node.

Mathematically represented, a decision tree is a disjunction (that is, union) of conjunctions (that is, intersections) of conditions on variables in data. Every path from the root to a leaf node is the conjunction of tests at the internal nodes, whereas paths to distinct leaf nodes are such that any observation can match from the root to a leaf node. The two measures of magnitude of a decision tree are the depth and breadth. The largest number of edges from the root to a leaf node is called the depth of tree. The number of nodes, whether internal or leaf nodes, at any particular level are breadth of tree at the specific level. The average numbers of edges from the root to leaf nodes arethe average depth of tree. The average number of nodes at different levels is the average breadth of tree.It is important to keep in mind, that there is no uniqueness in the decision tree grown from a given data set. Different optimality criteria can lead to different decision trees. The simplest decision trees are binary decision trees, and growing the minimal binary tree is an NP-complete problem. There are two major approaches of decision trees, namely the top-down and bottom-up. The literature shows a clear preference for the top-down approach (Rokach, L. and Maimon, O. (2005),D. Malerba et al.(2004), Ron Kohavi and Chia-Hsin Li(1995)), whereas very small amount of literature in the bottom-up approach(R.C. Barros et al.(2013), Erhun Kundakcioglu , Tongu ÇUnluyurt(2007)). Both approaches include prominent decision tree induction algorithms, which are reviewed in the following sections.

II.TOP-DOWN APPROACH TO INDUCTION

The foundations of the top-down approach to decision tree induction are provided by the Concept Learning System (CLS) framework of Hunt et al (1966). The objective of CLS is to minimize the cost of classifying data element. The concept of cost has two alternative interpretations. First, the measurement cost for a specified attribute of a data element. Second, the cost of classifying data element to class j when it really belongs to class k. Either way, the top-down approach leads to a recursive algorithm that restricts attention to the available decisions at the current depth of tree and attempt to minimize the cost in the restricted space before moving to one level down in the tree. The algorithm can be defined in terms of the following two steps. The two steps are implemented cyclically one after another by using the following notation.

The dataset at node t is denoted by X_t. The k classes are assigned the k labels in the set

$$Y = \{y_1, y_2, \ldots\ldots., y_k\}.$$

Step 1. If all observations X_t at node t have a common label y_t, then node t is made leaf node and is assigned the label y_t.

Step 2. If an observation Xt at node t has more than one label, then node t is divided into subsets by applying a test on one of the observed variables and observations are assigned to subsets according to result of the test.

Step 1 is the termination criterion and, the step 2 is implemented recursively until the termination criterion is satisfied.

It is obvious that this algorithm is oversimplified for practical use. It is the basis of all top-down decision tree induction algorithms. More specifically, the step 1 of this algorithm is too strong and may lead to non-termination of algorithm unless every individual observation from a leaf node. Also, this algorithm can work only if all possible combinations of values of the variables occur in the training data set and the training dataset do not have any inconsistency, which has a unique class label for all observations which have a common combination of values of variables. The termination criterion of the step 1 requires every pure leaf node. This would cause treeenormously in size and also lead to the problem of over fitting. It has, therefore, been proposed in the literature that the termination criterion can be relaxed by allowing a small level of impurity at a leaf node, so that, there is no risk of overgrowing the tree. Another proposal appears in literature is construction of a cost-based function for partitioning in a specified node. Some of the popular algorithms like ID3 and C4:5 use results from information theory to construct functions for partitioning nodes to the maximum possible extent.

Formally, the purpose of growingdecision tree is to identify a function for mapping all possible combinations of values of input variables into a predefined set of class labels, which can contain two or more labels. The training dataset is denoted by D the set of input variables is denoted by X, and the set of class labels is denoted by y = $\{y_1, y_2, \ldots, y_k\}$. The objective is to develop a classifier that minimizes the generalization error. The generalization error is defined as the misclassification rate. When the input variables are categorical, but not ordinal, their state space can be partitioned arbitrarily, while the state spaces of quantitative or numerical variables are partitioned in intervals, mostly in two intervals of form of half real lines. The universal instance space, denoted by U, is the Cartesian product of the state space X, the input variables and the state space y of the response variable. The training set is a finite subset of the universal instance space U and it consists of observations on inputvariables as well as the response variable. If the training data set is of size m, then it contains m tuples of the form $\{(x_1, y_1), (x_2,y_2), \ldots. (x_m, y_m)\}$ where $x_i \epsilon x$ and $y_i \epsilon$ y, i=1, 2, …., m. In order to obtain an optimal classification, it is necessary to define a criterion of optimality. Such a criterion is based on the generalized error that is sought to be minimized. The generalization error is defined in terms of the loss function and the distribution of D. It is defined as follows:

$$E(x,y) = \sum_{x \in X, y \in Y} L(x,y)\, D(x,y),$$

Where the loss function is defined by

$$L(x,y) = 1 \text{ if } y \neq I(x),$$

0 if y = I(x)

Where I(x) is the indicator function that checks if the data x associated with the response of y. In case of numerical variables, the summation is replaced by integration. It is obvious that the magnitude of error depends on the size of the training dataset.

Another important parameter related to a decision tree is its complexity. Tree complexity can be defined in terms of the following quantities.

- Total number of nodes
- Number of leaves
- Depth of the tree
- Number of distinct variables used in tree induction.

It must also be kept in mind that the decision tree induction can also be used as a way to rule induction. The tests at internal nodes leading to any particular leaf node constitute an antecedent. The class of the leaf node is the outcome or response. This one to one correspondence is one of the fundamental results in the decision theory.

III. NODE PARTITIONING CRITERIA

Node partitioning criteria has historically been univariate in the sense that they are based on the variable at time. It is, therefore, necessary to find the variable that best partitions the given node. The node is actually partitioned only if the best partition satisfies the partitioning criterion. If the best variable does not satisfy this criterion, then obviously no other variable can satisfy it and the node is not partitioned. However, induction of the tree is heavily influenced by the criterion used for partitioning of nodes, and it is, therefore, necessary to note the different criterion available for this purpose.

3.1 Criteria based on impurity

If a random variable takes k distinct values with probabilities $P = (p_1, p_2, \ldots. ,p_k)$, then an impurity function $\Phi : [0, 1]^k \rightarrow R$ is defined so as to have the following properties.

- $\Phi(P) \geq 0$
- $\Phi(P)$ is minimum when pi = 1 for at least one i = 1,2, … , k.
- $\Phi(P)$ is symmetric in its arguments
- $\Phi(P)$ is differentiable.

Note that, when only one pi = 1, then all other components of P are zero and the variable is said to be pure. On the other hand, the level of impurity is maximum when all components of P are equal for training dataset of S size probability distribution of the response variable y having k distinct values are given by

$$P(y) = (N_1/N, N_2/N, \ldots., N_k/N),$$

Where (N_1, N_2, …., N_k)is the vector of frequencies of the k distinct values of y, so that N = ∑ Ni. Now, suppose the data set S is partitioned according to a variable X that takes m distinct values, (x_1, x_2, …. x_ml, say and N_{ij} denotes the number of instances inS that have Y = y_i and X = x_j , then obviously.

$$\text{Ni} = \sum_{j=1}^{m} Nij$$

The reduction in impurity due to partitioning of S in m subsets according to the variable X is then given by

$$\Delta\Phi(X,S) = \Phi\ (P) - \sum_{j=1}^{m} \frac{Nij}{N} \Phi\ (Pj)$$

Where Pj = (P_{1j}, P_{2j} …., P_{kj}) is the conditional probability distribution of y in the subset of S given X = x_j

Information theoretic approach to the same situation uses anentropy function defined by

$$\text{Entropy (y)} = \sum_{i=1}^{k} \frac{Ni}{N} \log_2 \left(\frac{Ni}{N} \right),$$ and defines information gain due to partitioning S according to the variable X as

$$\text{Information Gain (X)} = \text{Entropy (Y)-} \sum_{j=1}^{m} \frac{N_{.j}}{N} Entropy\left(Y / X = x_j\right),$$

Where $N_{.j}$ is the number of observationsin S where X = x_j, j = 1,2, ….., m.

3.2Gini Index

Gini index is used as a measure of dispersion and is commonly used in economics as measurement of an inequality. It is defined by

$$\text{Gini (Y)} = 1 - \sum_{i=1}^{k} \left(\frac{Ni}{N} \right)^2,$$

The partitioning criterion for evaluating the variable X is then the gain in the Gini index as follows.

$$\text{Gain Gini (X)} = \text{Gini (Y)-} \sum_{j=1}^{m} \frac{N_{.j}}{N} Gini\,(Y / X = x_j),$$

3.3 Gini Ratio

Gain ratio is a normalized measure based on information gain relative to entropy and defined as follows

$$\text{Gain Ratio (X)} = \frac{Informatio\,n\,Gain\,(X)}{Entropy\,(X)}$$

The problem with gain ratio is that it is not defined when the denominator is zero. The other problem is that this ratio gets inflated when the denominator is very small. As a remedy, it is usually applied in the two stages. The first stage is to remove variables that have the information gain below the average values over all variables, so that, the gain ratio is reasonable at the second stage.

3.4 Normalized Impurity Measure

This measure normalizes an impurity measure as follows:

$$\mathrm{NI(X)} = \frac{\Delta\phi\,(\mathrm{x})}{-\sum_{i=1}^{k}\ \sum_{j=1}^{k} p_{ij}.\log_2 p_{ij}}$$

Where $p_{ij} = N_{ij} / N$, i = 1, 2, …, k, j = 1,2, … , m.

IV.Binary Partitioning Criteria

Partitioning criteria can be simplified when it is decided to grow a binary tree. A binary tree is a tree where every internal node is partitioned into two child nodes. Suppose the space S at an internal node is to be divided into two mutually exclusive and exhaustive subspaces D_1 and D_2, so that

$S = D_1UD_2$ and $D_1 \cap D_2 = \Phi$.

If the size of a set S is denoted by Na binary criterion known as the twoing criterion defined as follows.

$$\text{twoing } (D_1) = \frac{1}{4}\frac{N(D_1)\,N(D_2)}{[N(s)]^2}\left[\sum_{i=1}^{k}\ \frac{N(Y=Yi\,\&\,D_1)}{N(D_1)},\frac{N(Y=Y\,i\,\&\,D_2)}{N(D_2)}\right]$$

4.1 Angular Distance Criterion

A partitioning criterion based on the angular distance between the two vectors represents the probability distributions of Y over D_1 and D_2**,** also known as an orthogonality criterion defined as follows:

$AD\ (D_1) = 1 – COS\ \theta\ (P_1, P_2)$,

Where $\theta\ (P_1, P_2)$ is the angle, between P_1 and P_2, the probability distributions of Y over D_1 and D_2, respectively.

4.2 Kolmogorov – Smirnov Criterion

This criterion is used when the response variable Y is binary and takes two values y_1 and y_2. The criterion is defined by

$$KS\ (D_1) = P\ (Y = Y_1/D_1) – P\ (Y = Y_2/D_1).$$

V.Optimal Partitioning Strategy :

The objective of constructing a decision tree is to construct the tree optimally, so that, the resulting decision rules are optimal. The decision tree induction is a multi-step process, the most common practice to partition in a non-leaf node optimally until no node can be partitioned without any gain in information or reduction in impurity. As a result, there is no unique way of partitioning an internal node optimally. The best strategy, therefore, is to select an optimality criterion before partitioning any node. If information gain is the criterion, then the optimal partitioning rule maximizes the gain in information.In any case, a common consideration can be regarded the principle of parsimony, while the tree is being optimized by using any one of the optimality criteria mentioned above. It is true that there is no uniqueness in decision tree induction, and variations are inevitable. Nevertheless, the focus should be on the performance rather than the structure of the resulting decision tree.

VI. Conclusions

Every node splitting criterion splits an internal node optimally to generate child nodes until a decision tree is formed. It has been observed that distinct node splitting criteria need not lead to distinct child nodes or distinct decision trees. This observation implies that a split cannot characterize the node splitting criterion. Nevertheless, every decision tree is optimal according to at least one node splitting criterion. It is therefore recommended that decision tree induction is better if a single split satisfies multiple node splitting criteria. This essentially means that even though a decision tree is induced according to a specific node splitting criterion, it is desirable to verify if the resulting splits also satisfy any other node splitting criteria. The larger the number of node splitting criteria satisfied by the actual splits, better is the resulting decision tree.

REFERENCES

[1] D. Malerba et al., Top-down induction of model trees with regression and splitting nodes. IEEE Trans. Pattern Anal. Mach. Intell. 26(5), 612–625 (2004)

[2] Erhun Kundakcioglu ; TonguÇ UnluyurtBottom-Up Construction of Minimum-Cost and/ or Trees for Sequential Fault Diagnosis IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans (Volume: 37 , Issue: 5 , pp. 621 – 629, Sept. 2007)

[3] Farris, F. A. (2010). The Gini Index and Measures of Inequality. The American Mathematical Monthly, vol. 117, no. 10, pp. 851-864.

[4] Hunt, E. B., Marin, J., and Stone, P. J. (1966). Experiments in Induction. Academic Press, New York.

[5] R.C. Barros et al., A framework for bottom-up induction of decision trees, Neurocomputing (2013 in press)

[6] Rokach, L. and Maimon, O. (2005). Top-Down Induction of Decision Trees Classifiers - A Survey. IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, vol. 35, no. 4, pp. 476-487.

[7] Rokach, L. and Maimon, O. (eds.) (2010). Data Mining and Knowledge Discovery Handbook, 2nd Edition. Springer.

[8] Ron Kohavi and Chia-Hsin Li, Oblivious Decision Trees, Graphs, and Top-Down Pruning Appears in the International Joint Conference on Artificial Intelligence (IJCAI), 1995.