International Journal of Academic Engineering Research (IJAER)
ISSN: 2643-9085
Vol. 8 Issue 6 June - 2024, Pages: 6-9

Decision Tree Algorithms

Phd Shamiljon Rustamov1 and Norboyev Bexzod2

1Dean of the Faculty of Computer Engineering,
Karshi branch of the Tashkent University of Information Technologies
shamiljon1988@gmail.com
2Master's 1st stage student,
Karshi branch of the Tashkent University of Information Technologies
norboyevbexzod98@gmail.com

***Abstract**: Decision trees are a type of machine-learning algorithm that can be used for both classification and regression tasks. They work by learning simple decision rules inferred from the data features. These rules can then be used to predict the value of the target variable for new data samples. Decision trees are represented as tree structures, where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents a prediction. The algorithm works by recursively splitting the data into smaller and smaller subsets based on the feature values. At each node, the algorithm chooses the feature that best splits the data into groups with different target values.*

Keywords: ID3, C4.5, dataset, CART, CHAID, MARS, Versatile, Overfitting

INTRODUCTION

A flexible and comprehensible machine learning approach for classification and regression applications is the decision tree. The conclusion, such as a class label for classification or a numerical value for regression, is represented by each leaf node in the tree-like structure that is constructed, with each internal node representing a judgment or test on a feature.

To divide the data into subsets that are as pure as possible about the target variable, the tree is built recursively, beginning at the root node and selecting the most informative characteristic. The aforementioned procedure persists until a halting condition is fulfilled, generally at attaining a specific depth or upon the node possessing a minimum quantity of data points. Decision trees are a good tool for elucidating the logic behind forecasts since they are simple to see and comprehend.

They are prone to overfitting, though, which results in unduly complicated trees. Pruning methods are employed to lessen this. Moreover, decision trees provide the foundation for ensemble techniques that aggregate many trees to increase prediction accuracy, such as Random Forests and Gradient Boosting. In conclusion, decision trees are an essential machine learning tool that is appreciated for their versatility, interpretability, and ease of use.

Components of a Decision Tree

Before we dive into the types of Decision Tree Algorithms, we need to know about the following important terms:

- Root Node: It is the topmost node in the tree, which represents the complete dataset. It is the starting point of the decision-making process.
- Internal Node: A node that symbolizes a choice regarding an input feature. Branching off of internal nodes connects them to leaf nodes or other internal nodes.
- Leaf/Terminal Node: A node without any child nodes that indicates a class label or a numerical value.
- Parent Node: The node that divides into one or more child nodes.
- Child Node: The nodes that emerge when a parent node is split.

Working of the Decision Tree Algorithm

Whether employed for regression or classification, a decision tree method provides a flexible and easily interpreted machine learning technique. To create choices depending on the input features, it constructs a structure like a tree. Leaf nodes in the tree indicate the ultimate results, whereas nodes in the tree represent decisions or tests on the feature values.

Here’s a detailed breakdown of how the decision tree algorithm works:

- With all the data at its starting point, the process is the root node. In order to effectively divide the data into discrete classes or values, the algorithm chooses a feature together with a threshold. Depending on the job (classification or regression), the feature and threshold are selected to maximize information gain or decrease impurity.
- Depending on the outcome of the feature test, the data is separated into subgroups. When a characteristic like "Age" is used with a threshold of 30, for instance, the data is divided into two subsets: records with Age less than or equal to 30, and records with Age more than 30.
- For every subgroup, the splitting procedure is repeated, resulting in child nodes. Up until a halting condition is satisfied, this recursive process keeps going. A minimal amount of data points in a node, a predetermined tree depth, or the lack of additional information gained from splits beyond that point are examples of common stopping criteria.
- A node turns into a leaf node when a stopping requirement is satisfied. The final judgment or forecast is represented by the leaf nodes. Each leaf node is classified using the class label that is most common inside the subset. In a regression, the target variable's mean or median value within the subset is usually found in the leaf node.
- The tree structure that is produced can be understood. The reasoning of the model can be intuitively understood by viewing a decision path from the root to a leaf node as a set of rules.

Understanding the Key Mathematical Concepts Behind Decision Trees

To comprehend decision trees fully, it's essential to delve into the underlying mathematical concepts that drive their decision-making process. At the heart of decision trees lie two fundamental metrics: entropy and Gini impurity. These metrics measure the impurity or disorder within a dataset and are pivotal in determining the optimal feature for splitting the data.

Entropy:Entropy, denoted by H(D) for a dataset D, measures its impurity or disorder. In the context of decision trees, entropy represents the uncertainty associated with the class labels of the data points. If a dataset is perfectly pure (all data points belong to the same class), the entropy is 0. If the classes are evenly distributed, the entropy is at its maximum.

Mathematically, entropy is calculated using the formula:

$$H(D)=\sum_{i=1}^{n} p_i \log_2 p_i$$

Where p_i represents the proportion of data points belonging to class i in the dataset D. The base 2 logarithm is used to calculate entropy, resulting in entropy values measured in bits.

Information Gain:Information gain is a metric used to determine the effectiveness of a feature in reducing entropy. It quantifies the reduction in uncertainty (entropy) achieved by splitting the data based on a specific feature. Features with higher information gain are preferred for node splitting in decision trees.

Mathematically, information gain is calculated as follows:

$$Information\ Gain=H(D)-\sum_{v=1}^{V} \frac{|D_v|}{|D|}\mathrm{H}(D_v)$$

Where V is the number of values (unique outcomes) the feature can take, D_v represents the subset of data points for which the feature has the **v**th value, and |D| denotes the total number of data points in dataset D.

Gini Impurity: Gini impurity, often used in algorithms like CART (Classification and Regression Trees), measures the probability of misclassifying a randomly chosen element if it were randomly labeled according to the class distribution in the dataset. Gini impurity is computationally efficient and works well for binary splits.

Mathematically, Gini impurity for a dataset D is calculated as:

$$Gini(D)=1-\sum_{i=1}^{n} p_i^2$$

Where pi represents the proportion of data points belonging to class i in dataset D. Lower Gini impurity values indicate a purer dataset.

An approach for decision trees called ID3 (Iterative Dichotomiser 3) is employed in classification applications. It is one of the first and most used decision tree algorithms, created by Ross Quinlan in 1986. The ID3 algorithm builds a decision tree from a given dataset using a greedy, top-down methodology.

It works by greedily choosing the feature that maximizes the information gain at each node. ID3 calculates entropy and information gain for each feature and selects the feature with the highest information gain for splitting.

ID3 uses entropy to measure the uncertainty or disorder in a dataset. Entropy, denoted by H(D) for dataset D, is calculated using the formula:

$$H(D)=\sum_{i=1}^{n} p_i)\log_2(p_i)$$

Information gain quantifies the reduction in entropy achieved by splitting the data based on a particular feature. Features with higher information gain are preferred for splitting. Information gain is calculated as follows:

$$Information\ Gain=H(D)\text{-}\sum_{v=1}^{V}\frac{|D_v|}{|D|}\mathrm{H}(D_v)$$

Every decision tree node's dataset is recursively divided using the ID3 algorithm according to the chosen attribute. This method keeps going until either there are no more attributes to divide on, or all the examples in a node belong to the same class.

The decision tree may be trimmed after it is constructed in order to enhance generalization and lessen overfitting. In order to do this, nodes that do not considerably improve the correctness of the tree must be removed.

A couple of the ID3 algorithm's drawbacks are that it tends to overfit the training set and cannot directly handle continuous attributes. Owing to these drawbacks, other decision tree algorithms that address some of these problems have been developed, including C4.5 and CART.

Entropy, information gain, and recursive partitioning are three key principles in the ID3 algorithm, which is a fundamental technique for creating decision trees. Mastering these ideas is crucial to learning about decision tree algorithms in machine learning.

As an enhancement to the ID3 algorithm, Ross Quinlan created the decision tree algorithm C4.5. In machine learning and data mining applications, it is a well-liked approach for creating decision trees. Certain drawbacks of the ID3 algorithm are addressed in C4.5, including its incapacity to deal with continuous characteristics and propensity to overfit the training set.

A modification of information gain known as the gain ratio is used to address the bias towards qualities with many values. It is computed by dividing the information gain by the intrinsic information, which is a measurement of the quantity of data required to characterize an attribute's values.

$$GainRatio=\frac{Split\ gain}{Gain\ information}$$

Where Split Information represents the entropy of the feature itself. The feature with the highest gain ratio is chosen for splitting.

When dealing with continuous attributes, C4.5 sorts the attribute's values first, and then chooses the midpoint between each pair of adjacent values as a potential split point. Next, it determines which split point has the largest value by calculating the information gain or gain ratio for each.

By turning every path from the root to a leaf into a rule, C4.5 can also produce rules from the decision tree. Predictions based on fresh data can be generated using the rules.

C4.5 is an effective technique for creating decision trees that can produce rules from the tree and handle both discrete and continuous attributes. The model's accuracy is increased and overfitting is prevented by its utilization of gain ratio and decreased error pruning. Nevertheless, it might still be susceptible to noisy data and might not function effectively on datasets with a lot of features.

CONCLUSION

In summary, decision tree algorithms are a fundamental machine learning technique with broad applicability and several advantages. While they have limitations such as the risk of overfitting, these can be effectively managed with proper techniques and ensemble methods. Their interpretability and versatility make them a valuable tool for both exploratory data analysis and predictive modeling.

REFERENCES

1. Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

2. Quinlan, J. R. Norboyev B.U.Rustamov Sh.X(1986). *Induction of Decision Trees*. Machine Learning, 1(1), 81-106. doi:10.1023/A:1022643204877

3. Breiman, L. Norboyev B.U.Rustamov Sh.X (2001). *Random Forests*. Machine Learning, 45(1), 5-32. doi:10.1023/A:1010933404324