decision tree

Authors

Alexander Jung*, Department of Computer Science, Aalto University, Espoo, Finland
*Corresponding author: alex.jung@aalto.fi

Abstract

A decision tree is a flowchart-like representation of a hypothesis map: a root node reads in the feature vector of a data point and evaluates a basic test on it. Depending on the test result, the feature vector is forwarded to one of the child nodes. The child nodes apply further tests and forward the feature vector accordingly. This forwarding continues until the feature vector reaches a leaf node. A leaf node has no children and represents one specific function value, i.e., a prediction. The tree partitions the feature space into decision regions, one per leaf node, on which the represented hypothesis is constant. Decision trees serve as base models of the random forest and of gradient-boosted decision tree (GBDT) methods. Small decision trees yield interpretable machine learning (interpretable ML) if the user can comprehend the tests executed at their nodes.

Definition

A bank's loan approval can be written as a chain of simple tests on the applicant's features that ends in an "approve" or "reject" prediction. For example, is the income above a threshold? Is the credit score large enough? A decision tree is such a chain in general form: a flowchart-like representation of a hypothesis map h.

More formally, a decision tree is a directed acyclic graph (DAG) containing a root node that reads in the feature vector $\mathbf{x}$ of a data point. The root node then forwards the data point to one of its child nodes based on some elementary test on the features $\mathbf{x}$. If the receiving child node is not a leaf node, i.e., it has child nodes itself, it represents another test. Based on the test result, the data point is forwarded to one of its descendants. This testing and forwarding of the data point is continued until the data point ends up in a leaf node without any children, which carries the prediction (Fig. 1). The leaf nodes partition the feature space into decision regions; in Fig. 1, each test compares an entry of $\mathbf{x}$ with a threshold, so the decision regions are rectangles. Next to each node, Fig. 1 highlights the part of the feature space that contains the feature vectors reaching that node.

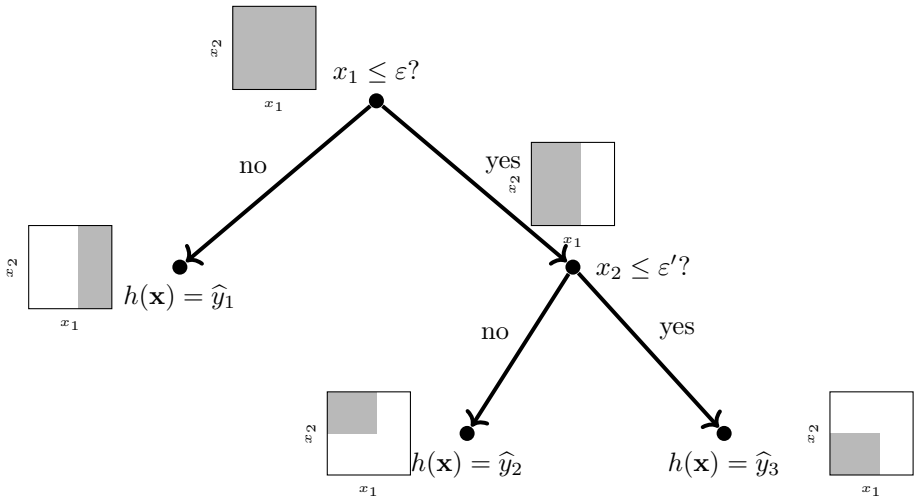

Figure 1: Decision tree as a flowchart-like representation of a piecewise constant hypothesis $h : \mathcal{X} \to \mathbb{R}$ for feature vectors $\mathbf{x} = (x_1, x_2)^\top$. Each test compares a single feature with one of the thresholds $\varepsilon, \varepsilon' > 0$. Next to each node, the shaded part of the feature space (depicted as a square) contains the feature vectors that reach that node. The shaded parts at the leaf nodes are the decision regions $\mathcal{R}_{\widehat{y}} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \widehat{y}\}$: rectangles, since every test involves a single feature

Training a decision tree amounts to extending a leaf node by attaching another decision tree. In the simplest case, a leaf node is replaced with a test node whose children are new leaf nodes (Fig. 2).

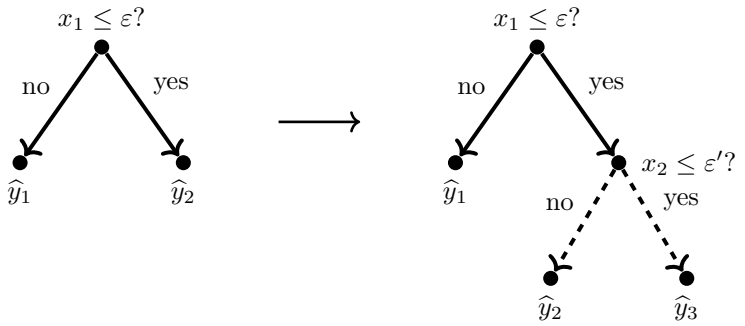

Figure 2: Training grows a decision tree. The tree on the right is obtained from the tree on the left by replacing the leaf node with prediction $\widehat{y}_2$ with a test node whose children are two new leaf nodes (dashed edges)

Widely used training methods for decision trees include CART (Breiman et al., 1984), ID3 (Quinlan, 1986), and C4.5 (Quinlan, 1993). Starting from a single leaf node, CART repeatedly replaces a leaf node with a test node: among candidate tests that compare a single feature with a threshold, it greedily selects the one that most reduces an impurity measure of the label values routed to the new children. Common impurity measures are the Gini index and the entropy for classification, and the variance for regression (Hastie et al., 2009, Sect. 9.2). This selection is a combinatorial search over candidate tests rather than a gradient-based update of continuous model parameters. The growth stops when a leaf node contains too few data points or no candidate test reduces the impurity substantially (Hastie et al., 2009, Sect. 9.2.2). Fig. 3 shows the result of this training on real data: each day of 2024 at the Finnish Meteorological Institute (FMI) weather station Helsinki Kaisaniemi is a data point, with the minimum temperature of the day as its feature and the maximum temperature as its label. The learned depth-2 tree is a piecewise constant hypothesis with four pieces, one per leaf node.

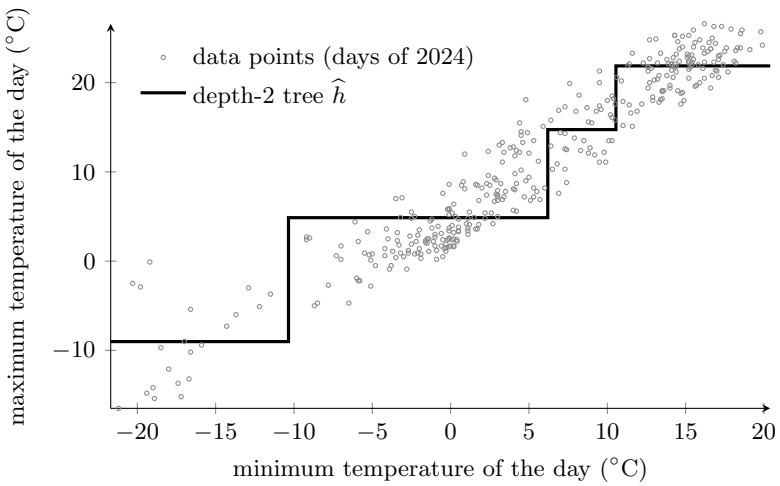

Figure 3: Scatterplot of the 366 days of 2024 at the FMI station Helsinki Kaisaniemi, with the minimum temperature of a day as its feature and the maximum temperature as its label. The step function is the piecewise constant hypothesis $\widehat{h}$ learned by a depth-2 regression tree: the three thresholds of its test nodes split the feature axis into four intervals, and on each interval the tree predicts the mean label of the data points routed to that leaf node. Data generated by `pythondemos/decisiontree.py`

Decision trees serve as base models of the random forest and of gradient-boosted decision tree (GBDT) methods, and their explicit tests make each prediction traceable. A decision tree with a small number of nodes is often consid-

ered interpretable: its computation can be traced by following the data point from the root node to a leaf node (Molnar, 2025). This makes small decision trees a common choice in interpretable machine learning (interpretable ML).

Cross References

decision region, classification, random forest, GBDT, feature importance, interpretable ML

References

1. Breiman et al. (1984). Classification and Regression Trees. Wadsworth, Belmont, CA.

2. Molnar (2025). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. 3rd ed., Ebook.

3. Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA.

4. Quinlan (1986). Induction of decision trees. Machine Learning, 1(1), pp. 81–106.

5. Hastie et al. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed., Springer Science+Business Media, New York, NY, USA.