Introduction to Machine Learning

Decision Trees

Varun Chandola

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
chandola@buffalo.edu

University at Buffalo
Department of Computer Science and Engineering
School of Engeering and Applied Sciences

Outline

Why Decision Trees?

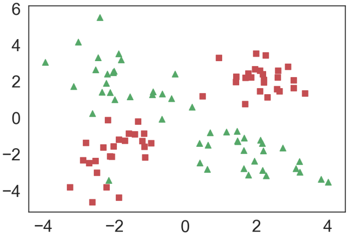
- Linear models are easy to interpret/explain but have limited power
- Non-linear models can be more accurate but are “black-boxes”

Why do we care about interpretability and explainability?

- Builds trust, transparency, and accountability into the model
- Needed for fairness and ethical considerations of ML

Decision Trees

- Inherently "non-linear" model

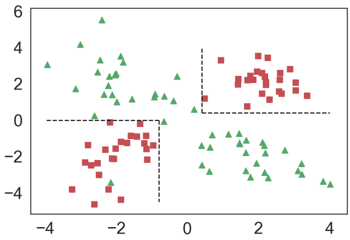


- No linear boundary
- Divide the region ($\mathcal{X}$) into non-intersecting sub-regions

$$\begin{aligned} \mathcal{X} &= \cup_{i=0}^{n} R_i \\ \text{s.t. } R_i \cap R_j &= \varnothing, \text{for } i \neq j \end{aligned}$$

Decision Trees

- Inherently "non-linear" model



- No linear boundary
- Divide the region ($\mathcal{X}$) into non-intersecting sub-regions

$$
\begin{aligned}
\mathcal{X} &= \cup_{i=0}^{n} R_i \\
\text{s.t. } R_i \cap R_j &= \varnothing, \text{for } i \neq j
\end{aligned}
$$

How to select regions

- Computationally intractable
- Decision trees - approximate solution via a greedy, top-down, recursive partitioning scheme.
- Start with $\mathcal{X}$ and split it into two child regions by thresholding on a single feature
- Continue splitting nodes using a feature and a threshold
- Formally, given a parent region R_p, a feature index j, and a threshold $t \in \mathbb{R}$, we obtain two child regions as:

$$\begin{aligned} R_{p1} &= \{\mathbf{x} | x_j < t, \mathbf{x} \in R_p\} \\ R_{p2} &= \{\mathbf{x} | x_j \geq t, \mathbf{x} \in R_p\} \end{aligned}$$

How to choose the splits?

- Need a loss function $L()$ as a set function on a region R
- For a given parent R_p, we can calculate the decrease in loss as:

$$\delta = L(R_p) - \frac{|R_1|L(R_1) + |R_2|L(R_2)}{|R_1| + |R_2|}$$

Cross-entropy Loss

$$L_{cross}(R) = -\sum_{c} \hat{p}_c \log_2 \hat{p}_c$$

- $\hat{p}_c$ is the probability of observing an example of class c in the given node

$$\hat{p}_c = \frac{|\mathbf{x} : class(\mathbf{x}) = c, \mathbf{x} \in R|}{|R|}$$

- If $\hat{p}_c = 0$ then $\hat{p} \log_2 \hat{p} \equiv 0$

Alternatives Cross-entropy Loss

Gini Index/Loss

$$L_{gini}(R) = 1 - \sum_{c} \hat{p}_c^2$$

Other Considerations

- Categorical features
- Regularization (pruning)
- Computational complexity - $O(N * D * d)$
  - N - number of training examples
  - D - number of features
  - d - depth of the tree

Variants of Decision Trees

- **Regression Trees** - Use a different loss function
- **Random Forests** - An ensemble of decision trees

References