Lecture 9 Decision Trees

wbg231

December 2022

introduction

- Decision trees are an inherently non linear type of model
- they are lso a good way to understand Ensemble methods.

Decision trees

- regression trees try to predict a continuous outcome
- classification trees try to predict a discrete class.
- a binary tree has 2 children nodes there are also multi-way trees that have more
- each node contains a subset of teh data
- the data splits created by each node involve only a single feature
- for continuous variables the splits are always of the form $x_i \leq y$
- for discrete variables we partition the values into two stets
- predictors are made in the leaf nodes

constructing the tree

- our goal is to find the boxes $R_1 \dots R_J$ that minimize $\Sigma_{j=1}^{J}\Sigma_{i\in R_j}(y_i - \hat{y}_{R_j})$ subject to complexity constraints
- the issue with this is finding the true optimal binary tree is computationally intractable
- so instead we use a greedy algorithm (that is we take the best choice at every step) where we start at the root and on our first step take the splits that would result in the minimal loss, and then pass the data split like that to the next step and have each for each of those sections pick the best splits and continue this until we hit some stopping criteria

- we only split regions defined in the last step at each current step we predict based on the mean value of a terminal node ie $\hat{y}_{R_m} = mean(y_i | x_i \in R_m)$
- so building a tree like this we are making the best local choice at every step but are unlikely to reach the overall optimal choice
- so the left is how the tree looks for regression
- each node, is a binary decision with a condition that splits the tree remaining data into 2 groups in the binary case
- the the right side, is the search space. as you can see node (that is condition) make s a linear subdivision of our search space.
- geometrically this looks like this kind of step wise division as can be seen in this chart.

finding the best split point

- there are infinitely many split points for each feature.
- suppose we have a vector $D = \{(x^1, y^1)...(x^n, y^n)\}$ where $x^i \in \mathbb{R}^d, y^i \in \mathbb{R} \forall i \in [1, n]$
- suppose we are not considering splitting on the jth feature. ie x^i_j
- we can sort our values by there value of the jth feature as $x_j(1) \cdots x_j(n)$
- lets only consider split points between two adjacent values. note that any split point between the two same values will have the same loss.
- due to this it is common to split half way between the two adjacent values

  $$s_j \in \{\frac{1}{2}(x_j(r) + x_j(r+1) | r = 1 \cdots n-1\}$$

  so this is just saying that geometrically we are taking our split to be halfway between two adjacent points

decision trees and over fitting

-