Tree-based methods partition the feature space into a set of rectanggles and fit a simple model in each one. They are conceptually simple yet powerful tools.

Given a region R which is a subset of domain D, define function I_R on D

$$I_R(p) = \begin{cases} 1 & \text{if } p \in R, \\ 0 & \text{if } p \notin R \end{cases}$$

Consider a regression problem with a collection of responses y and features $x \in \mathbb{R}^p$. A regression tree consists of a partition of $\mathbb{R}^p$ $R_1, R_2, \cdots, R_m$ and a prediction of y for each region in the partition: $c_1, c_2, \cdots, c_m$. Formally

$$\hat{y} = \sum_{i=1}^{m} c_i I_{R_i}(x)$$

We will recursively define the process of growing a tree. Suppose we already have a partition $R_1, \cdots, R_{m-1}$. On each of the region R_i, we want to further split the region into two parts: $R_{i1}^{js} = \{x | x_j < s, x \in R_i\}$ and $R_{i2}^{js} = \{x | x_j >= s, x \in R_i\}$. Such a split has two unfixed parameters: which feature we are going to split over (j) and where we are goint to split over (s). We select this value by achieving the best local result in the target function. Suppose we have mean-square cost fnction. Then

$$\begin{aligned} (j,s) &= \arg\min_{j,s}[\min_{c_1} \sum_{x_k \in R_{i1}^{js}} (y_k - c_1)^2 + \min_{c_2} \sum_{x_k \in R_{i2}^{js}} (y_k - c_2)^2] \\ &= \arg\min_{j,s}[\sum_{x_k \in R_{i1}^{js}} (y_k - \bar{y}_{i1}^{js})^2 + \sum_{x_k \in R_{i2}^{js}} (y_k - \bar{y}_{i2}^{js})^2] \end{aligned}$$

Here $\bar{y}$ is the average of y which belong to the region indicated by the subscripts of $\bar{y}$

Obviously a tree can overfit the data. We introduce a one regularization method. The idea is to first grow a relativelly large tree using primitive method, then prune it.

We grow the tree until the number of nodes reach a fixed number. Call this tree T_0. Consider any subtree $T \subset T_0$ which can be realized by collapsing some of T_0's nodes. Define cost function

$$C(T) = \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \bar{y}_m)^2 + \alpha |T|$$

$|T|$ is the number of leaves of T and $R_1, R_2, \cdots, R_m$ are the partition of $\mathbb{R}^p$ of the prediction model corresponding to T (the regions represented by the leaves of T). α is a tunning parameter of model.

For any α, we can find $T_\alpha = \arg\min C(T)$ through weakest link pruning: That is we succesively close the node of T_0 which produces the smallest increase pernode increase in $C(T) - \alpha|T|$ until we are left with one node. It can be shown that this sequence contains T_α.

Suppose we want to fit a certain model on a training data. Bagging or bootstrap aggregation fits the model on a collection of bootstrap samples and average their prediction result. Bagging reduces the variance and maintain the same bias as a single model would have.

Random forests is a modification of bagging that builds a large collection of de-correlated trees and average over them. The procedure can be described as:

1. Parameters: B, N, n_{max}, m
2. For b in $1 \dots B$
   (a) Draw a bootstrap sample of size N from the training data
   (b) Grow a tree T_b with node n_{max} with standard procedure with one modification: each time we split the node, we randomly pick m direction in feature space $\mathbb{R}^p$ and use them rather than using the whole feature space.
3. Make prediction by averaging over the result of all T_b