Random Forests

Classification and Regression Trees (CART)

Breiman et al. 1984

- Classification tree

  —-Categorical response

  —-Objective: to find a set of partitioning rules that classify observations that share a common category

- Regression tree

  —-Numerical response

  —-Objective: find rules that group observations that have a similar value of the response

Examples:

- Detecting the spam email messages based upon the message header and content; Categorizing animals as mammals and non-mammals upon the results of body temperature and giving birth or not; Categorizing cells as malignant or benign based upon the results of MRI scans etc,.
- Predict the mean ozone levels based on some independent variables
- Use credit card usage behaviors on over 25,000 customers of a large bank to predict:

  credit score is at least 720 or not (a classification problem)

  the actual credit score (a regression problem)

Classification tree of Mammals:

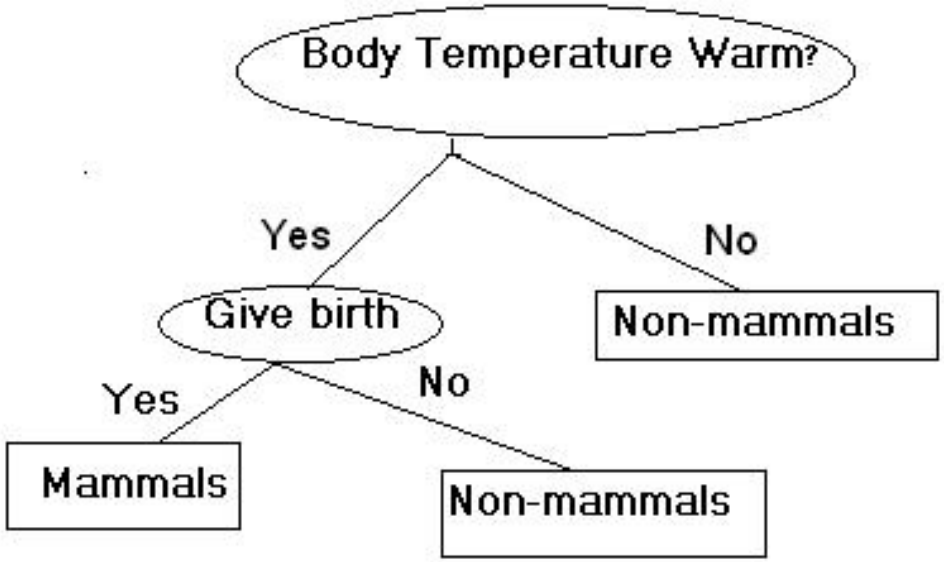
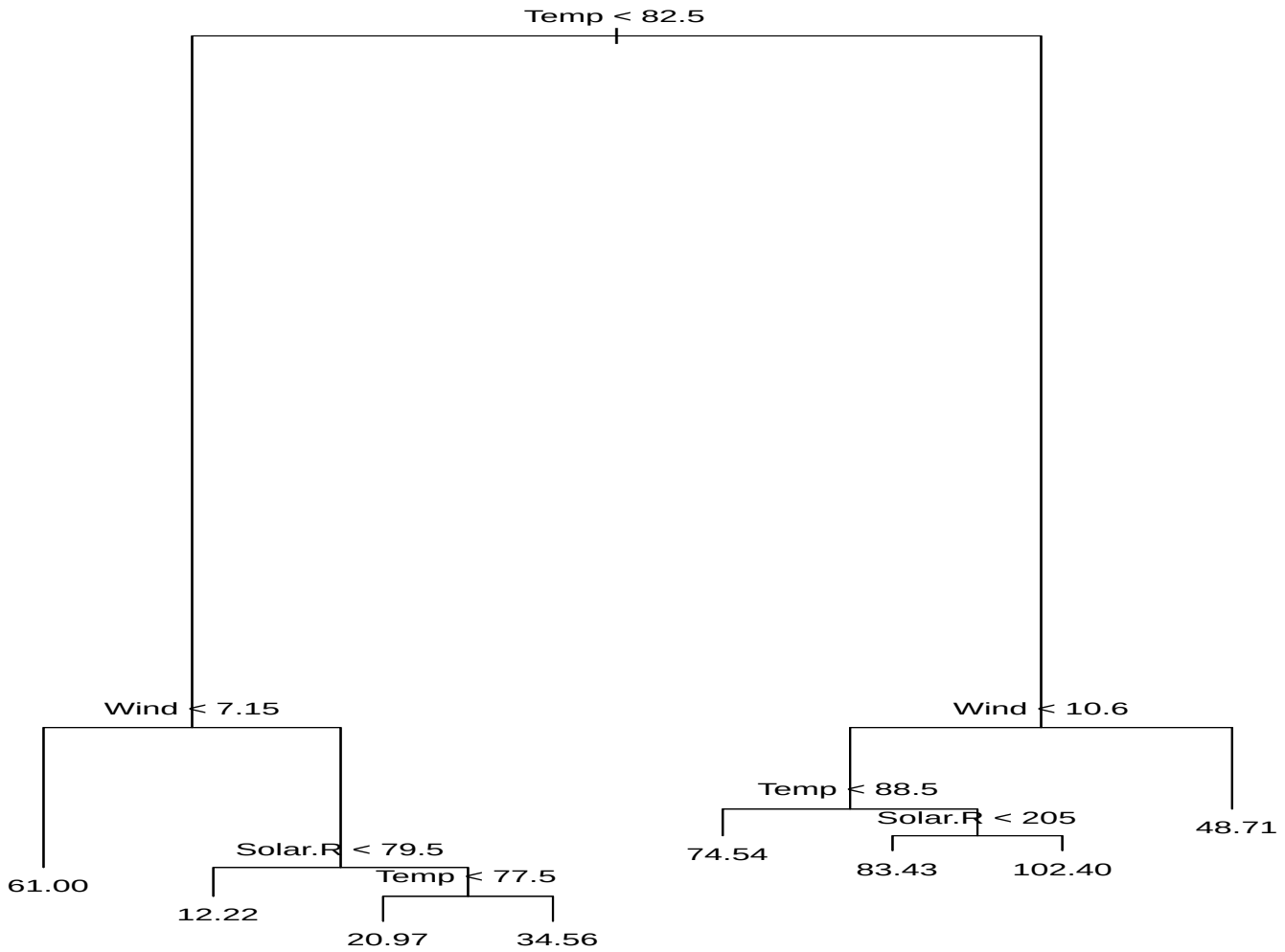


Figure 1: Regression tree of mean Ozone (ppb)



Some notation

- $\mathbf{x} = (x_1, x_2, \cdots, x_n)$, each $\mathbf{x}$ is assumed to belong to a predetermined class, as determined by the class label attribute $y = f(\mathbf{x})$
- The set of $\mathbf{x}$ used to construct the model is called a training set
- Root node: no incoming edges and zero or more outgoing edges
- Internal nodes: each of which has exactly one incoming edge and two or more outgoing edges
- Leaf or terminal nodes: each of which has exactly one incoming edge and no outgoing edges

Cart Algorithm: To partition the data into separate regions in which the response value is constant within each region

- The tree is built by the following process: First the single variable is found which best splits the data into two groups. The data is separated, and then this process is applied separately to each sub-group, and so on recursively until the subgroups either reach a minimum size or until no improvement can be made

  —select the best split at a node while building the tree

  —"pruning" to find the right-sized tree

—The CART algorithm attempts to determine the best variables for splitting and also the optimal split points to partition the feature space into disjoint regions

- The constructed model is used to classify future or unknown objects. To predict a response using a fitted tree model, the values of the predictors are used to determine simple binary conditions at each internal node. If the condition holds, then the left path is chosen, otherwise the right path is chosen. This continues down the tree model until a terminal node is reached and, hence, a prediction is made.

- Accuracy of the model is evaluated by percentage of test set samples that are correctly classified by the model

Regression Trees

- Partition the feature space into M regions: $R_1, R_2, \cdots, R_M$ with each region labeled with a constant response c_m
- Given $\mathbf{x}_0 \in R^p$ (suppose that we have p variables), predict a response using

$$\hat{f}(\mathbf{x}_0) = \sum_{m=1}^{M} c_m I(\mathbf{x}_0 \in R_m) \quad (1)$$

- Minimize $\sum (y_i - \hat{f}(\mathbf{x}_i))^2$ to get $c_m = \text{ave}(y_i | \mathbf{x}_i \in R_m)$
- The partitioning is done stepwise analyzing each variable separately at each step

- Let X_j be the jth variable and S_j be the set of distinct values of X_j in the data set
- The data is separated into two half-planes

  $$R_1(j,s) = \{\mathbf{x}|X_j \leq s\}, R_2(j,s) = \{\mathbf{x}|X_j > s\}, s \in S_j.$$

  $|S_j| - 1$ possible half plane sets in total
- Find the best splitting point for X_j that minimizes the sum of squared errors. Repeated for other variables to find the first split

- After finding the first optimal split, the data are partitioned and the process is continued on the two new subsets of data. This is repeated over and over again, until a minimal node size (say 5 observations) is reached or a node is homogeneous
- After the growing process is completed, each region in the final partition corresponds to a terminal node on the resulting tree model.

Advantages of CART algorithm

- Nonparametric approach, no parametric form required
- Interpretability, intuitive and simple
- Handling of complex relationships, identify nonhomogeneous relationships
- Scalability, an unlimited amount of predictors can be used to build a tree model, the predictors can be continuous or categorical with no major changes to the algorithm
- Robustness, highly robust with respect to outliers or misclassified points

Disadvantages of CART algorithm

- Modeling process is very data dependent
- Large tree, may overfit the data and will produce inaccurate predictions on future observations, prune the tree if necessary
- Lack of smoothness
- Instability, MAJOR DRAWBACK

  — relative instability compared to traditional statistical methods

  — small changes in the data can have dramatic changes in the final tree model

  — high instability is due to the variability in selecting the optimal splitting variable and/or its splitting point at each stage in the tree-building process

Tree-based ensemble methods

Bagging (Breiman, 1996): build a set of trees and get the average prediction

- Average the tree prediction to get a more stable prediction
- Bagging dramatically reduce the variability of unstable models such as those of CART and produces a smoother prediction surface

Randomization: Split at each node, is determined from a randomly chosen subset of all M predictors

- Randomize the internal decisions made by the base algorithm
- Produces a smoother prediction surface

Random Forests (RF) (Breiman, 2001)

——A cart ensemble method that combines both approaches of bagging and randomization

- Notations

  —number of observations N

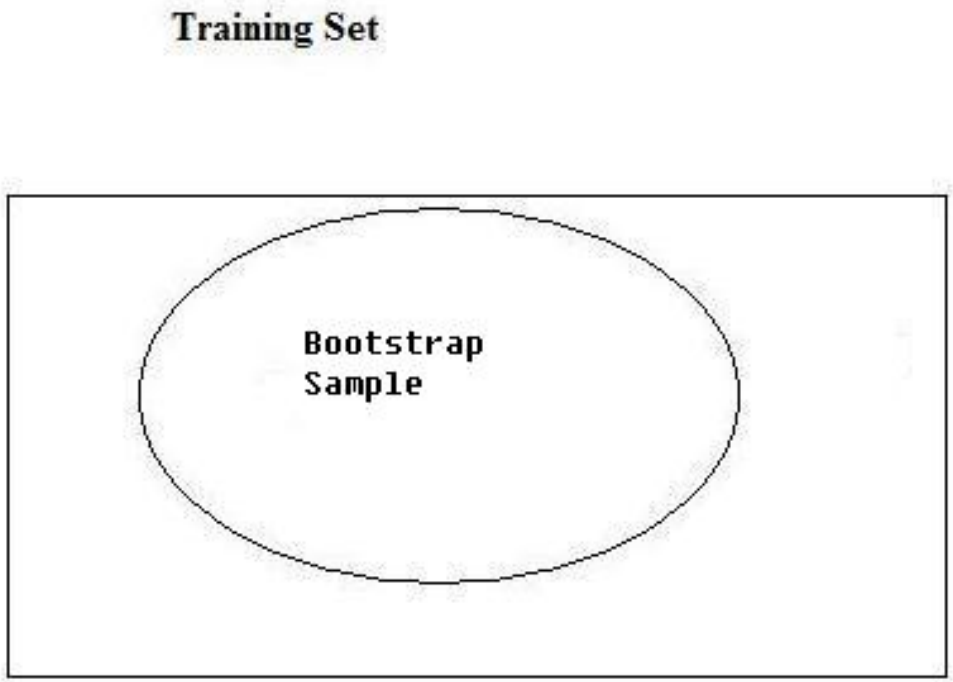
  —number of predictor variables M

  —training set $T = \{(y_i, \mathbf{x}_i), i = 1, 2, \cdots N\}$, $\mathbf{x}_i$: a vector of M variates

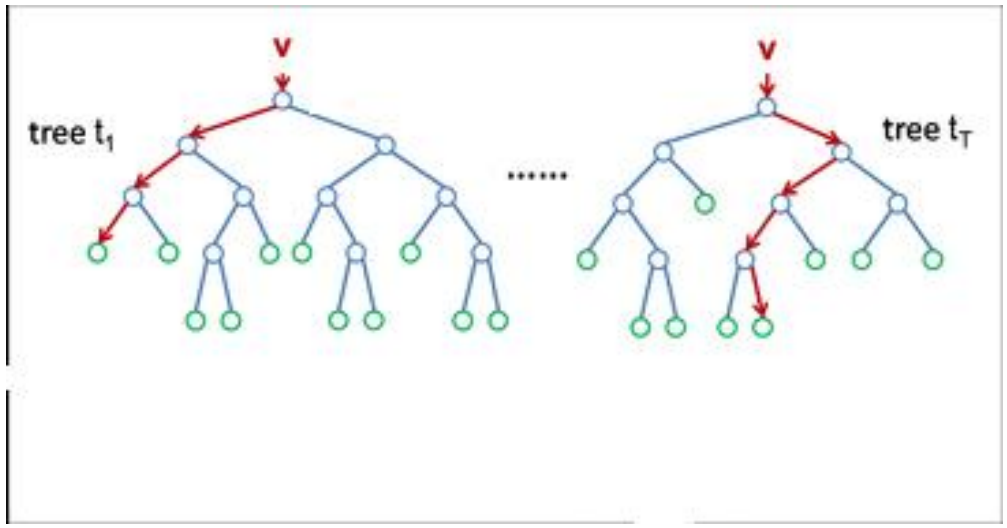
- Regression function

$$y_i = f(\mathbf{x}_i) + \epsilon_i \tag{2}$$

- Main goal: estimate the function f for the purpose of approximating y at future observations of $\mathbf{x}$



- Form bootstrap samples T_k, $k = 1, 2, \cdots, K$ of equal size of the training set T, each bootstrap sample is used to construct one tree, $T_k \longrightarrow \gamma_k$



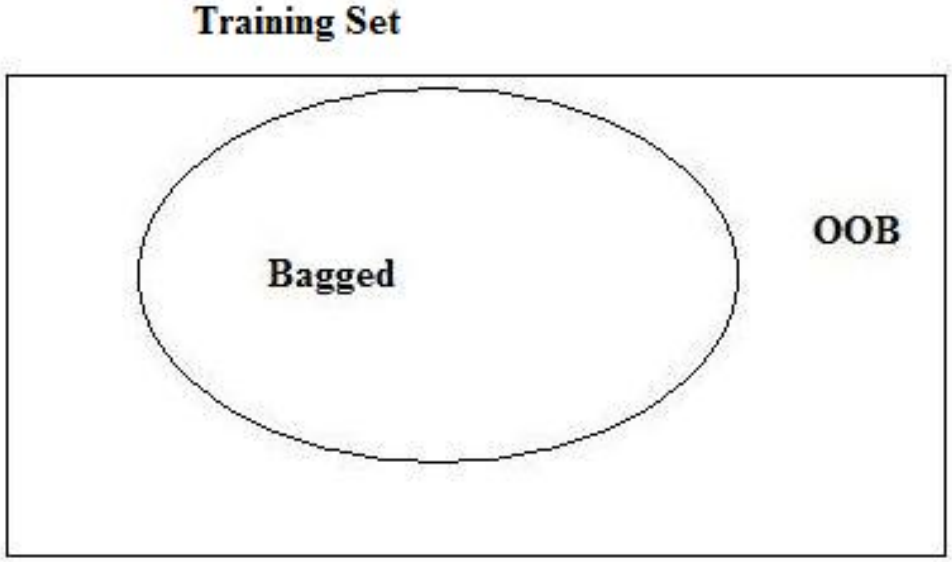
- Split at each node, is determined from a randomly chosen subset of all M predictor variables
- $\hat{f}(\mathbf{X}, \theta_k)$: prediction from the kth tree model in the forest
  ——$\mathbf{X}$: input vector
  ——θ_k: mechanism which creates tree γ_k

- RF prediction

$$\hat{f}_{RF}(\mathbf{X}_0, \Theta) = K^{-1} \sum_{k=1}^{K} \hat{f}(\mathbf{X}_0, \theta_k)$$

$\Theta = \{\theta_k\}_{k=1}^{K}$ is the set of realized random vectors

Out-of-bag (OOB) observations: the cases that are not selected (bagged) in the bootstrap samples



——T_k^{OOB}: for tree k, the portion of the bootstrap sample that is OOB

——k_i trees that do not use observation i during their construction

——averaging predictions at x_i over these trees to get the random forest OOB prediction

RF OOB prediction

$$\hat{f}^{OOB}(\mathbf{x}_i) = k_i^{-1} \sum_{k=1}^{K} \hat{f}_k(\mathbf{x}_i) I\left((y_i, \mathbf{x}_i) \in T_k^{OOB}\right)$$

Measure of effectiveness of Random Forests

- Prediction Error
- Variable importance scores
- Proximity measures

Properties of Random Forests

- Accuracy of RF has been shown to be competitive with many other data-mining techniques.
- Powerful in dealing with small sample size, high-dimensionality, and complexity data structures
- RF do not overfit the data as number of trees increase
- As more trees are added, a limiting value of the prediction error is achieved
- By randomly selecting a subset of predictors at each node, the correlation between trees is reduced while the strength of each tree is kept relatively constant

- Disadvantage: like a “black box” without much to say about the relationship between the response and explanatory variables ——RF model is a set of hundreds of trees which does not lend themselves easily to interpretation

R software

Random forests: R package randomForest (Liaw and Wiener, 2006)

References

Breiman L (1996) Bagging predictors. Machine learning 24:123–140

Breiman L (2001) Random forests. Machine learning 45:5–32

Liaw A, Wiener M (2006) Classification and regression by randomforest. R news 2:18–22