

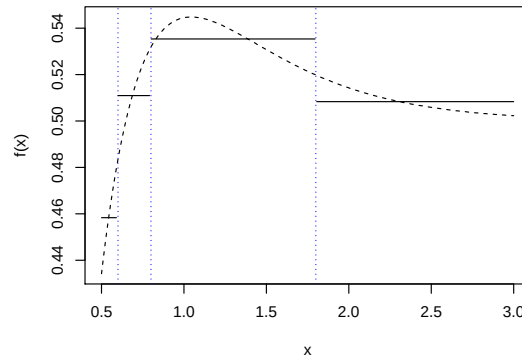
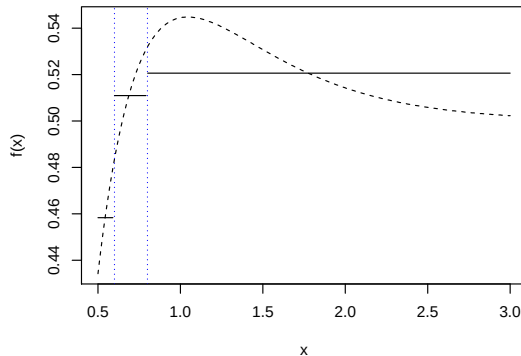
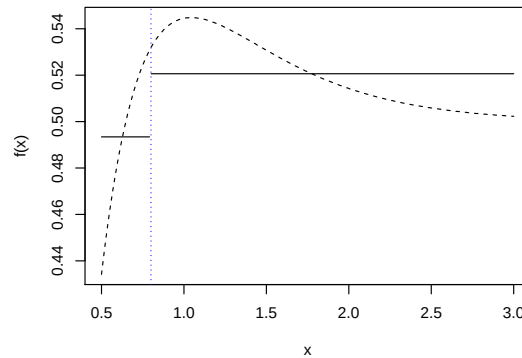
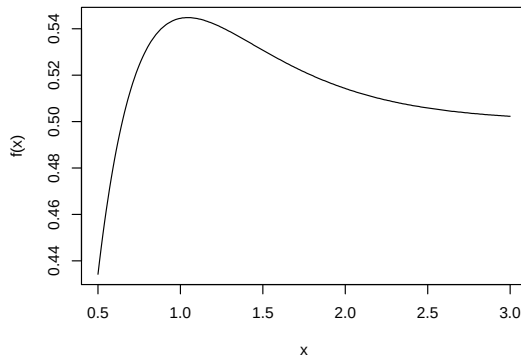
Classification and regression trees

Geir Storvik and Aliaksandr Hubin

May 6, 2022

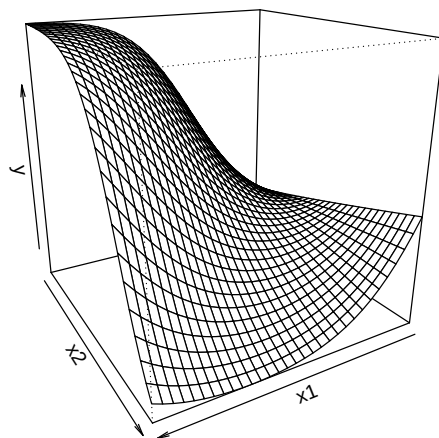
Regression trees

- Assume again division of x -axis into intervals
- Simplest assumption: Constant within each interval



- How many intervals?
- Placement of intervals?
- Values within intervals?

Two dimensions

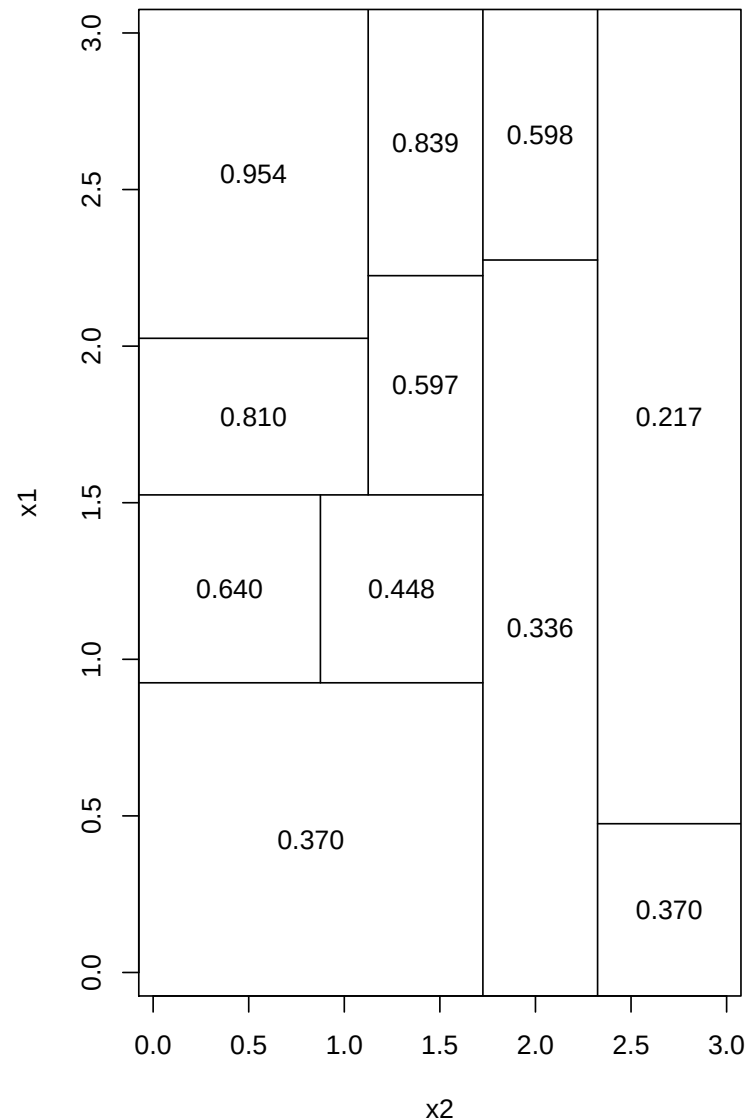
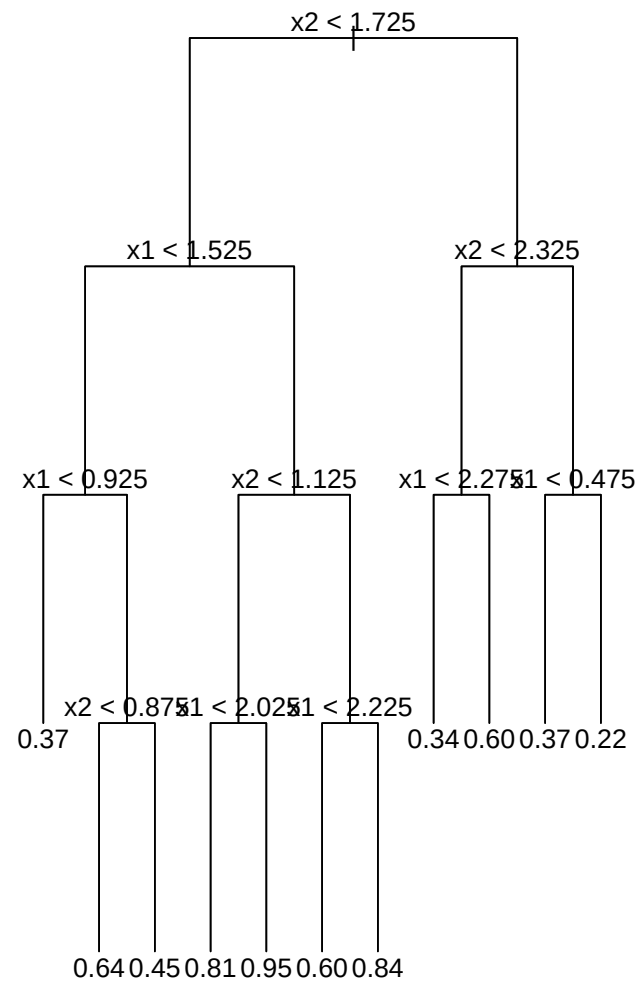


- Example of **Binary trees**
- General approximation

$$f(\mathbf{x}) = \sum_{h=1}^J c_h I(\mathbf{x} \in R_h)$$

- Region defined by sequential split of existing region **by thresholding wrt one covariate**

Tree structures



Fitting regression trees

- Fitted model $\hat{f}(x) = \sum_{h=1}^J \hat{c}_h I(x \in R_h)$
- Objective function:

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i))^2 \\ &= \sum_{h=1}^J \left\{ \sum_{i \in R_h} (y_i - \hat{c}_h)^2 \right\} = \sum_{h=1}^J D_h \end{aligned}$$

- Fitting:
 - Start with $J = 1$ and $\hat{f}(\mathbf{x}) = \bar{y}$
 - If R_h is divided into two parts, R'_h, R''_h , then

$$D_h \rightarrow D_h^* = \sum_{i \in R'_h} (y_i - \hat{c}_{h'})^2 + \sum_{i \in R''_h} (y_i - \hat{c}_{h''})^2$$

- with gain $g_h = D_h - D_h^*$
- Search through all **regions** R_h , all possible **variables** to separate on and all **separation points**, choose the one with largest gain
 - Stop when $J = n$ (or earlier, all **leaves** smaller than a given number)

Pruning

- Previous procedure: **Overfitting**
- Possible strategy: Stop earlier
- Better strategy: Make large tree and then **prune**!
- Introduce penalty or **cost-complexity**

$$C_{\alpha}(J) = \sum_{h=1}^J D_h + \alpha J$$

- Theoretical result: When we increase α , we can find optimal trees by a sequence of **pruning operations** on the current tree.
- Optimizing $C_{\alpha}(J)$: Sequentially remove leaf which minimizes $C_{\alpha}(J)$
 - The leaf which makes smallest increase in $\sum_{h=1}^J D_h$
- Choise of α :
 - AIC: $\alpha = 2\hat{\sigma}^2$
 - Cross-validation
- `Auto_tree.R`

Classification trees

- Assume $Y \in \{0, 1, \dots, K - 1\}$
- Following ideas from **regression trees**

$$\hat{p}_k(\mathbf{x}) = \sum_{j=1}^J P_{j,k} I(\mathbf{x} \in R_j)$$

where $P_{j,k}$ is the (estimated) probability for class k in region j .

- Main difference: Each leave now contain a **class probability**
- Can also do **hard classification**:

$$\hat{Y} = \arg \max_k \hat{p}_k(\mathbf{x})$$

Estimation

- Given regions R_j :

$$\hat{P}_{j,k} = \frac{1}{n_j} \sum_{i:\mathbf{x}_i \in R_j} I(y_i = k)$$

with $n_j = \sum_{i=1}^n I(\mathbf{x}_i \in R_j)$

- Also need to define R_j based on some **fitness** criterion
- For binary responses (2 classes):

$$\Pr(Y_i = y_i | \mathbf{x}_i) = p(\mathbf{x}_i)^{y_i} (1 - p(\mathbf{x}_i))^{1-y_i}$$

$$\log \Pr(Y_i = y_i | \mathbf{x}_i) = y_i \log p(\mathbf{x}_i) + (1 - y_i) \log(1 - p(\mathbf{x}_i))$$

- Assume now $\hat{p}(\mathbf{x}_i) = P_j$ for $\mathbf{x}_i \in R_j$. Define **deviance**

$$D = -2 \sum_{i=1}^n \{y_i \log \hat{p}(\mathbf{x}_i) + (1 - y_i) \log(1 - \hat{p}(\mathbf{x}_i))\}$$

- Specification of tree as for regression, but criteria for split of region now based on deviance

Deviance and entropy

- Assume $\hat{P}_j = \sum_{i:\mathbf{x}_i \in R_j} y_i / n_j$

$$\begin{aligned} D &= -2 \sum_{i=1}^n \{y_i \log \hat{p}(\mathbf{x}_i) + (1 - y_i) \log(1 - \hat{p}(\mathbf{x}_i))\} \\ &= -2 \sum_{j=1}^J \sum_{i:\mathbf{x}_i \in R_j} \{y_i \log \hat{P}_j + (1 - y_i) \log(1 - \hat{P}_j)\} \\ &= -2 \sum_{j=1}^J \{ \log \hat{P}_j \sum_{i:\mathbf{x}_i \in R_j} y_i + \log(1 - \hat{P}_j) \sum_{i:\mathbf{x}_i \in R_j} (1 - y_i) \} \\ &= -2 \sum_{j=1}^J n_j [\hat{P}_j \log \hat{P}_j + (1 - \hat{P}_j) \log(1 - \hat{P}_j)] \\ &= -2n \sum_{j=1}^J \frac{n_j}{n} [\hat{P}_j \log \hat{P}_j + n_j(1 - \hat{P}_j) \log(1 - \hat{P}_j)] = 2n \sum_j \frac{n_j}{n} Q(\hat{P}_j) \end{aligned}$$

with

$$Q(\hat{P}_j) = -\hat{P}_j \log \hat{P}_j - (1 - \hat{P}_j) \log(1 - \hat{P}_j) = - \sum_{k=0,1} \hat{P}_{j,k} \log \hat{P}_{j,k}$$

Entropies

and $\hat{P}_{j,1} = \hat{P}_j, \hat{P}_{j,0} = 1 - \hat{P}_j$

- Q also called **impurity** measure

Alternative (impurity) measures

- **Entropy:** $Q(\mathbf{P}) = \sum_k P_k \log(P_k)$
 - $Q(\mathbf{P}) \geq 0$, $Q((0, 1)) = Q((1, 0)) = 0$

- **Gini index**

$$Q(\mathbf{P}) = \sum_k P_k(1 - P_k)$$

- $Q(\mathbf{P}) \geq 0$, $Q((0, 1)) = Q((1, 0)) = 0$

- **Misclassification error**

$$\sum_{j=1}^J \frac{1}{n_j} \sum_{i \in R_j} I(y_i \neq \hat{y}_i)$$

- Start with $\hat{P}_k = \frac{1}{n} \sum_{i=1}^n I(y_i = k)$
- Search through all **regions** R_h , all possible **variables** to separate on and all **separation points**, choose the one with largest gain
- Stop when $J = n$ (or earlier, all **leaves** smaller than a given number)
- Illustration: `phoneme.R`

Tree methods - properties

- Advantages

- Logical simplicity and ease of "communication"
- Simple, compact mathematical formulation
- Speed of computation
- Can easily deal with categorical covariates
- Can include robust forms of deviance
- Variable selection

- Disadvantages

- Instability in results
- Difficulty in updating
- Statistical inference
- Selection of variables