

Trees

Statistical Learning

Master in Big Data. University of Santiago de Compostela

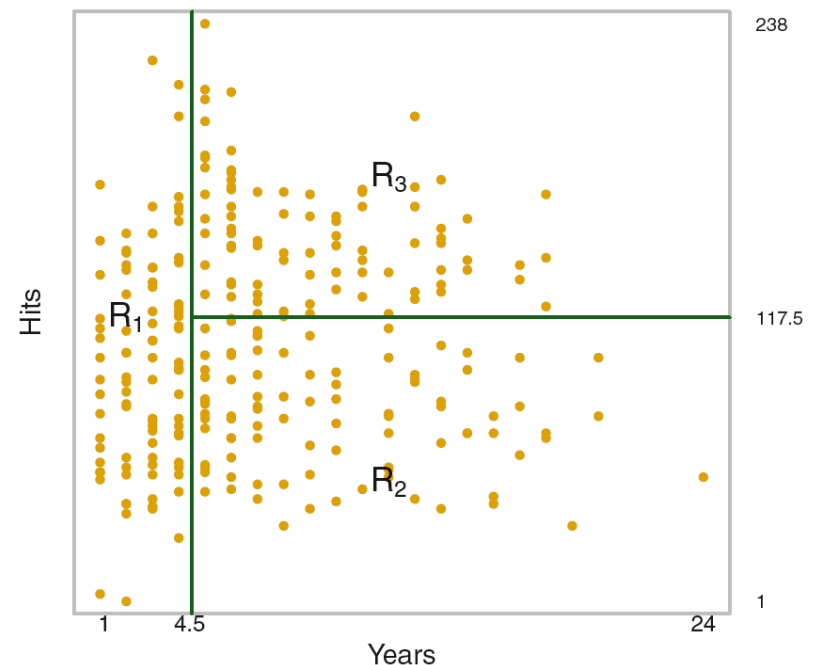
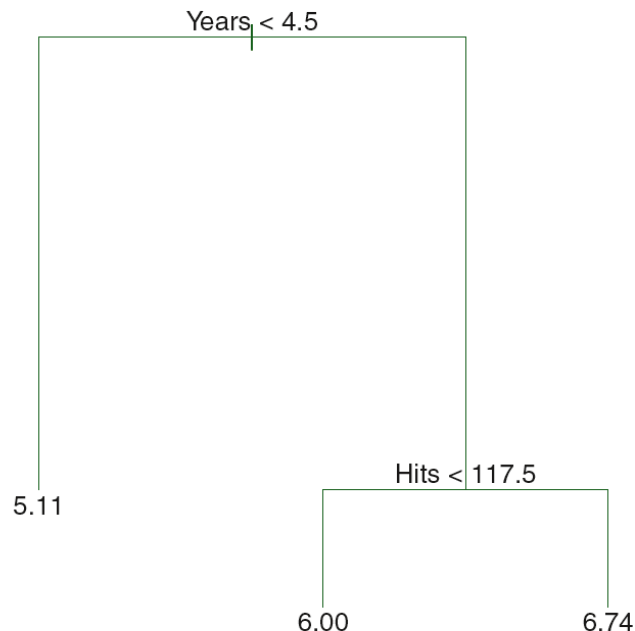
Manuel Mucientes

Background

- Stratify or segment the input space into a number of simple regions
- Splitting rules summarized in a decision tree
- Predictions of new observations: mean or mode of the training observations in the region
- Simple and useful for interpretation
- Not competitive with best supervised learning approaches
- Dramatic improvements when used in combination with bagging or boosting
- We focus the discussion on CART: classification and regression tree

Example

- Predicting the baseball player's salaries using regression trees
 - Inputs: years, hits, etc.
 - Output: salary (log-transformed)
 - R_1 mean log salary (\$1,000) of \$165,174 (5.107); R_2 \$402,834; R_3 \$845,346
 - Terminal nodes or leaves, internal nodes, branches
 - Easy to interpret, nice graphical representation



Regression Trees

- Two steps:
 - Divide the predictor space (set of possible values of X_1, \dots, X_p) into J distinct and non-overlapping regions, R_1, \dots, R_J
 - For each observation that falls in R_j , the prediction will be the mean of the output values of the training observations in R_j
- How do we construct the regions, R_1, \dots, R_J ?
 - Regions are boxes (high dimensional rectangles)
 - Find boxes that minimize RSS:
$$\sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$
 - Computationally infeasible to consider every partition into J boxes

Regression Trees (ii)

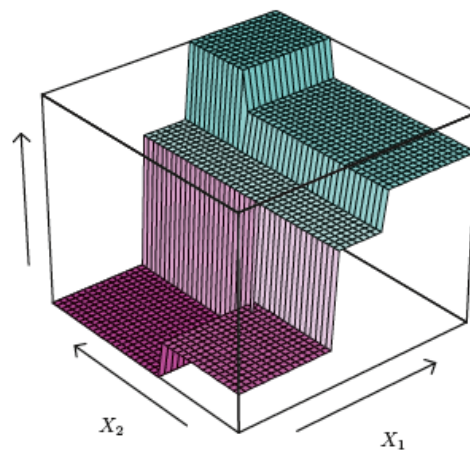
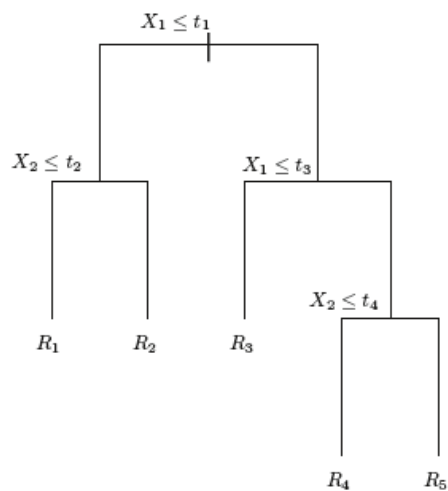
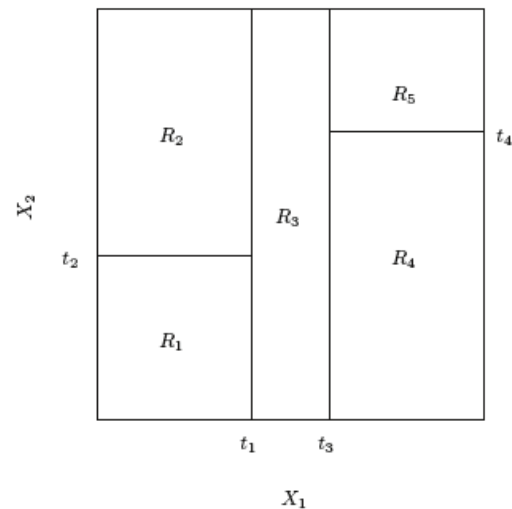
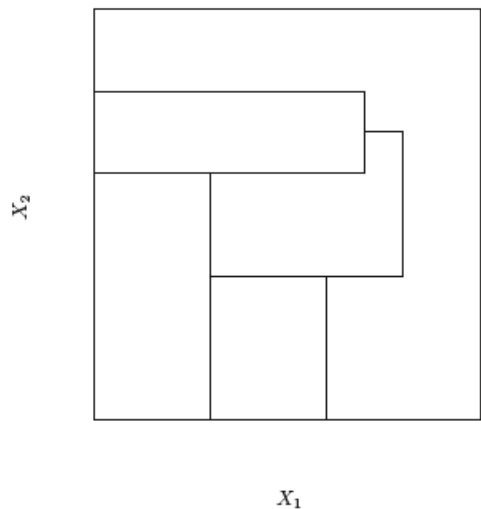
- Top-down greedy approach: recursive binary splitting
 - For each box, select the predictor X_j and the cutpoint s to minimize RSS within each region:

$$R_1(j, s) = \{X | X_j < s\} \quad \text{and} \quad R_2(j, s) = \{X | X_j \geq s\}$$

$$\sum_{i: x_i \in R_1(j, s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j, s)} (y_i - \hat{y}_{R_2})^2$$

- Continue until stopping criterion is reached: usually a minimum node size
- Predict the response of a new test observation using the mean of the training observations in the region
 - Predict the confidence: standard deviation

Example



Tree Pruning

- The previous method generates complex trees: overfitting
- Smaller tree: lower variance, better interpretation, slightly higher bias
- One possible solution: build the tree so long as the decrease in RSS due to a split is under some threshold
 - Smaller trees, but short sighted strategy: a worthless split may be followed by a very good split later on
- A better strategy: grow a large tree, and prune it back
 - Best way to prune?
 - Cross-validation: extremely large number of subtrees
 - Cost complexity pruning (weakest link pruning)
- We will use the minimum node size as stopping criterion

Cost Complexity Pruning

- Collapse the internal node that produces the smallest per-node increase in: $|T|$

$$\sum_{m=1}^{|T|} \sum_{i: x_i \in R_m} (y_i - \hat{y}_{R_m})^2$$

- Sequence of trees with different sizes

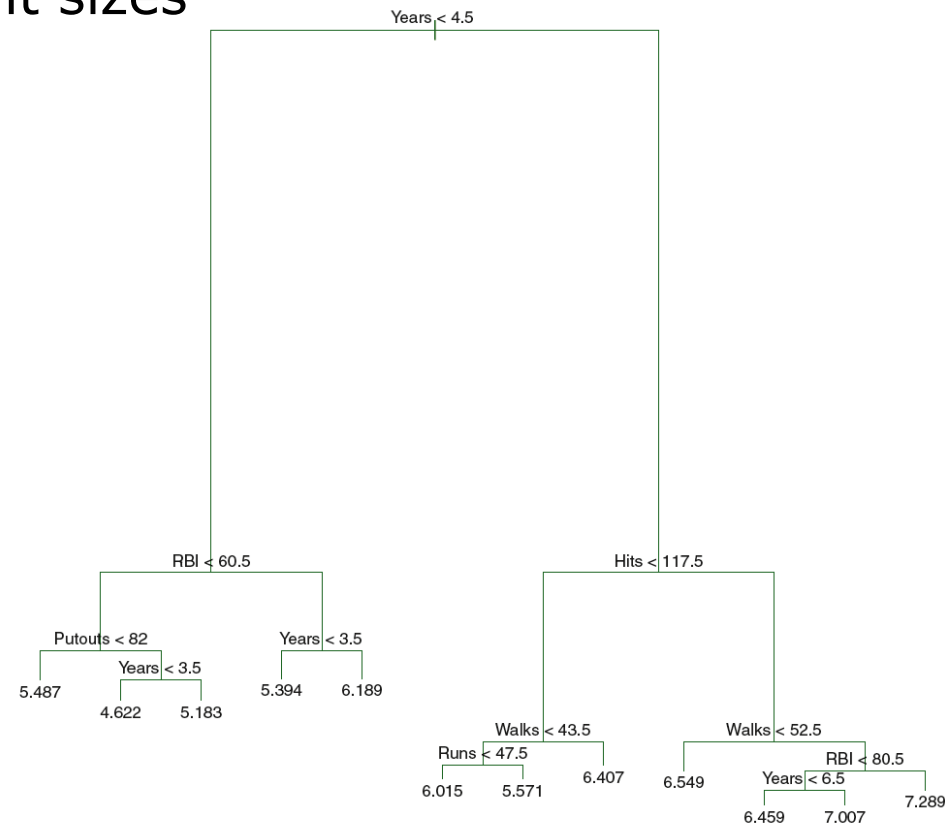
- Example: hitters data

- Tree learned with 9 features

- Training: 132 examples

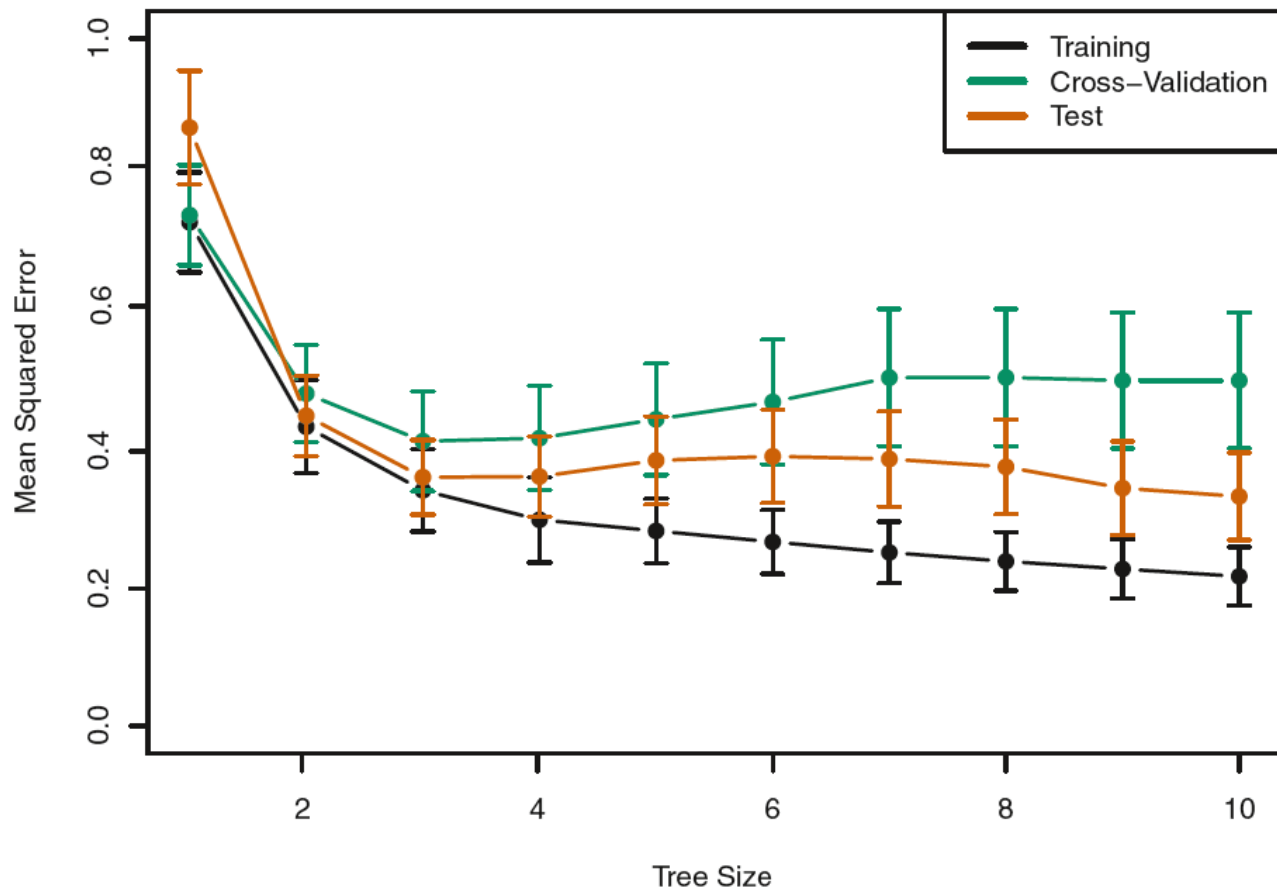
- Test: 131 examples

- Unpruned tree



Example (hitters data)

- Six-fold cross-validation (number of examples multiple of 6)
- Minimum error: three node tree



Classification Trees

- Similar to a regression tree, but with qualitative response
- Predicted class: most commonly occurring class of training observations in the region to which it belongs
 - We are also interested in the class proportions among the training observations in that region
- Tree growing: recursive binary splitting (as in regression)
 - RSS cannot be used
 - A natural alternative: classification error rate
 - Fraction of the training observations of a region that do not belong to the most common class
 - $E = 1 - \max_k(\hat{p}_{mk})$ for the m-th region
 - Not sufficiently sensitive for tree-growing

Classification Trees (ii)

- Two other measures are preferable

- Gini index:

- $$G = \sum_{k=1}^K \hat{p}_{mk}(1 - \hat{p}_{mk})$$

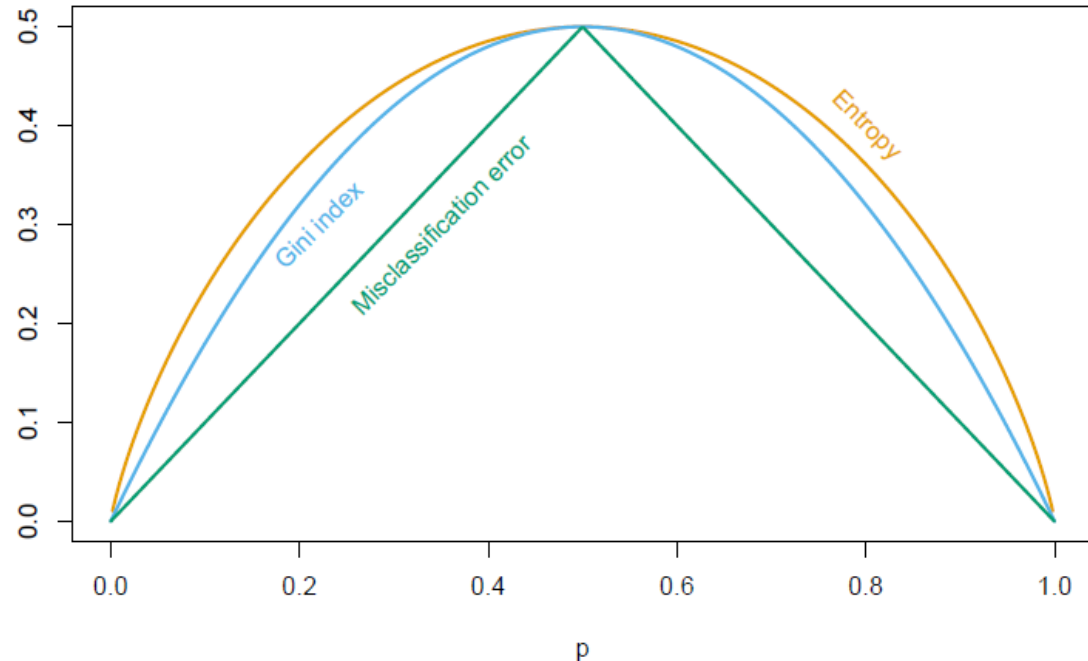
- Total variance across the K classes

- Cross-entropy:

- $$D = - \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$$

Node impurity measures for two-class classification

Cross-entropy has been scaled

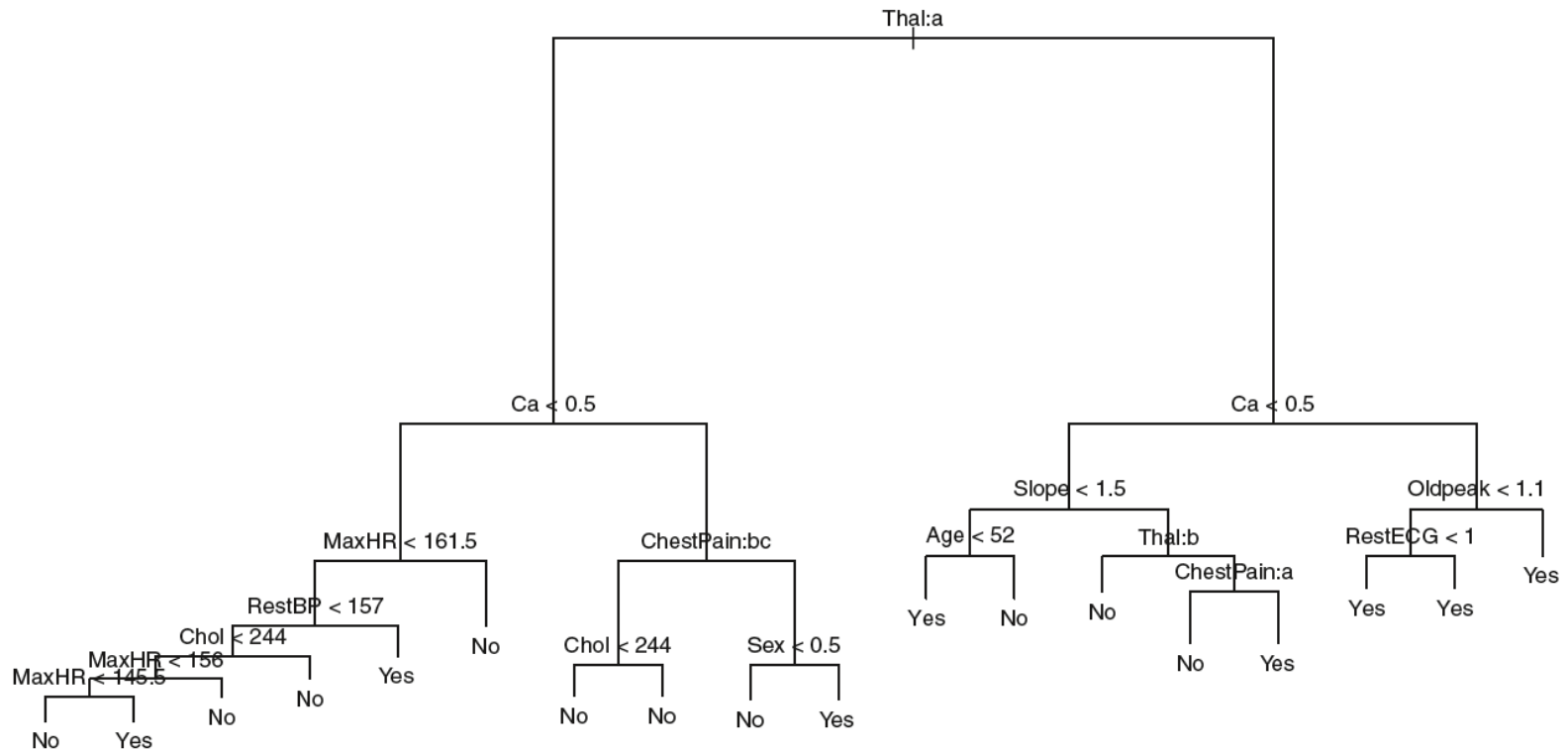


Classification Trees (iii)

- We need to weight the node impurity measures by the number of observations in the two created child nodes
- For building the tree:
 - Gini index and cross-entropy are preferable: more sensitive to node purity
- For pruning the tree:
 - Any of the three approaches might be used
 - Classification error rate is preferable if prediction accuracy of the final pruned tree is the goal

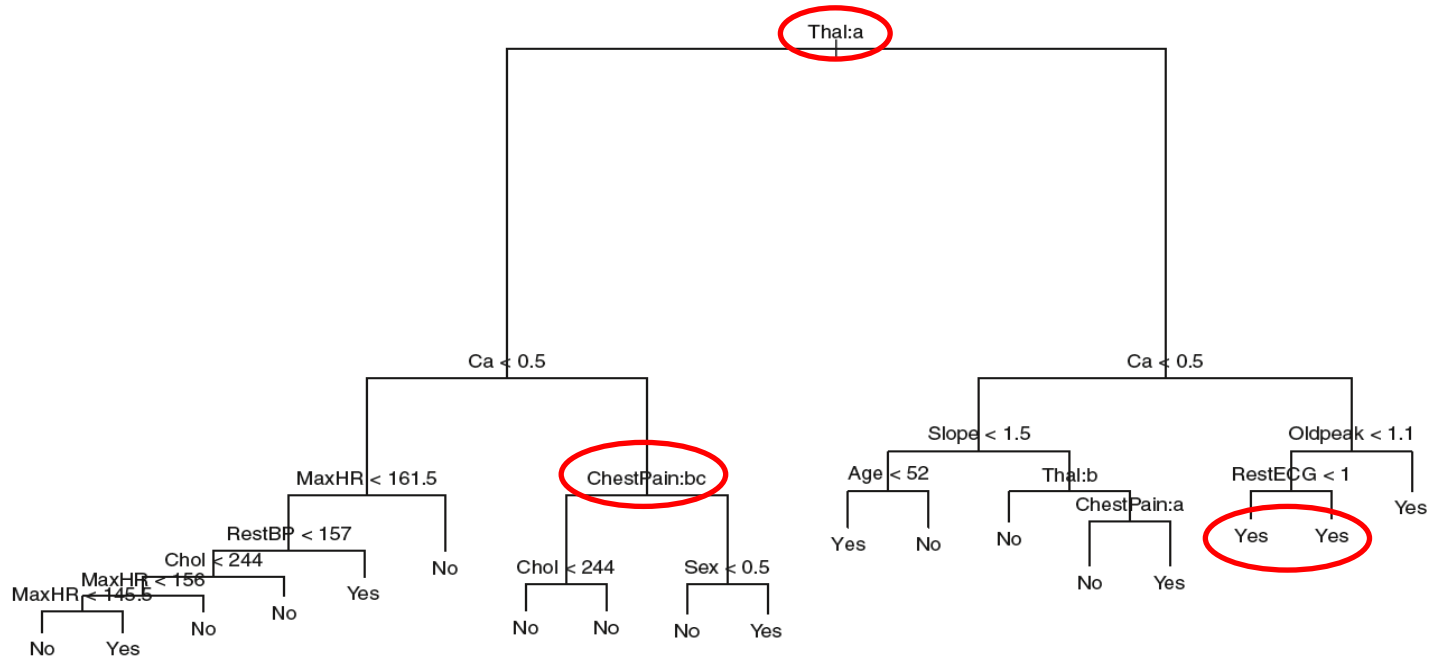
Example: Heart dataset

- Binary outcome (HD) for 303 patients
 - HD: Yes (heart disease) or No
- 13 predictors: Age, Sex, Chol (cholesterol measurement), Thal (Thalium stress test), ChestPain, ...
 - Categorical (qualitative) predictors: Sex, Thal, ChestPain



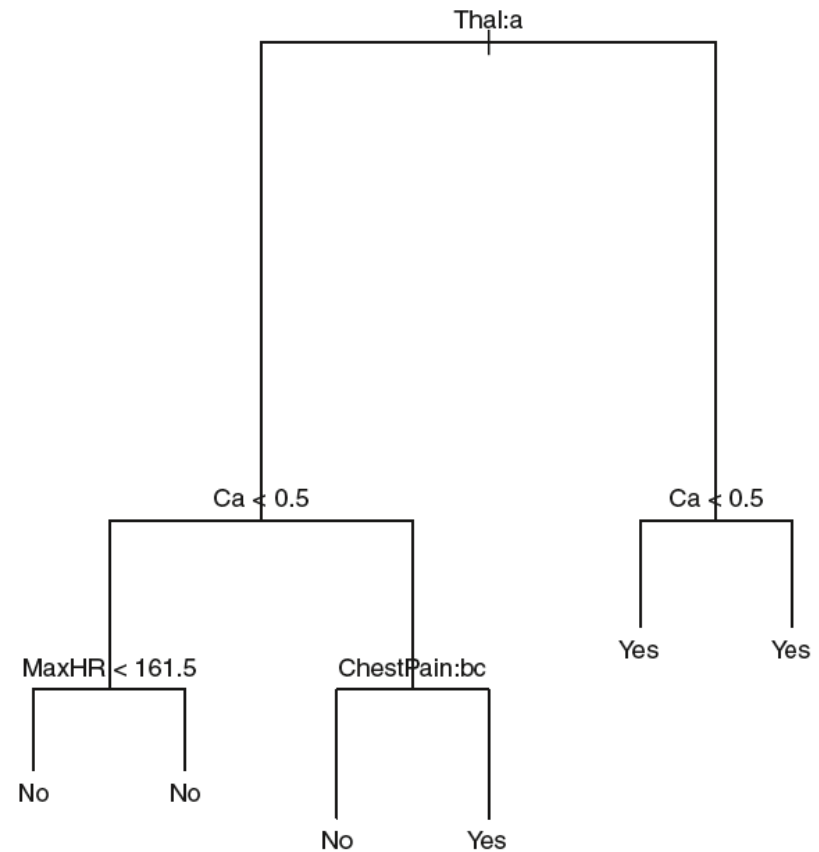
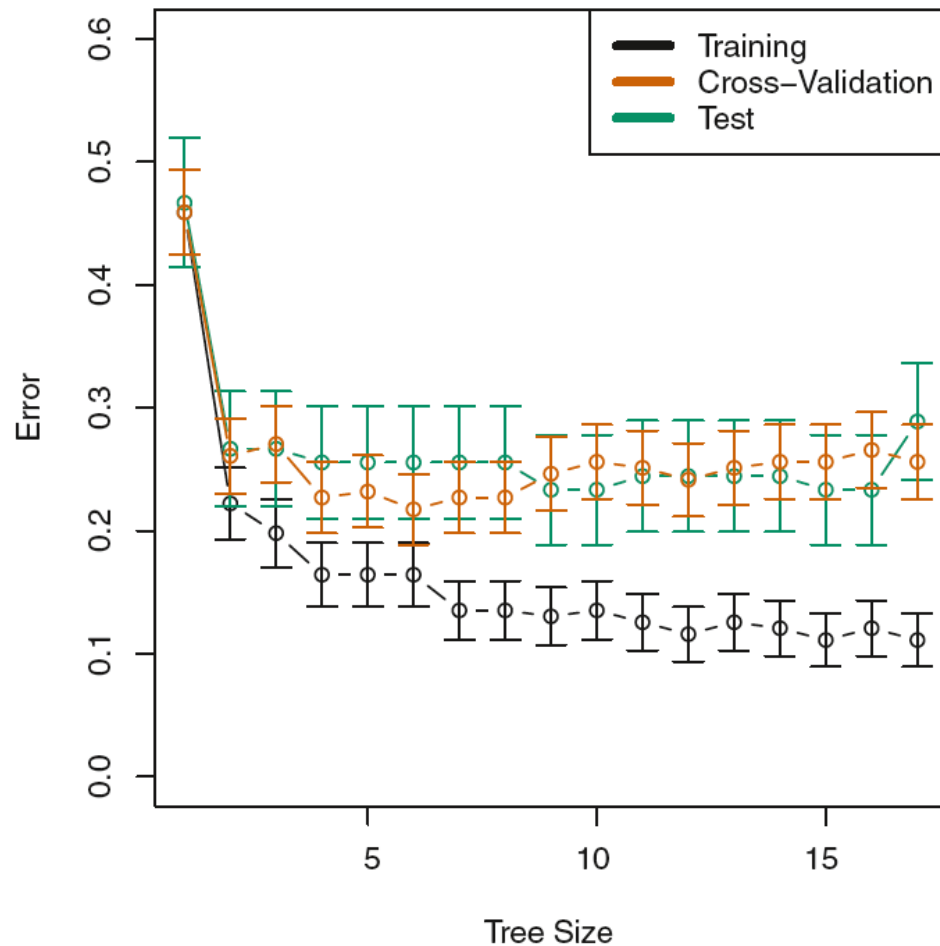
Example: Heart dataset (ii)

- ChestPain: (a) typical angina, (b) atypical angina, (c) non-anginal pain, (d) asymptomatic
- RestECG: increased node purity
 - Improves Gini index and cross-entropy
 - Classification error not improved
 - Right-hand leaf: 9/9 observations with response value Yes
 - Left-hand leaf: 7/11 observations with response value Yes



Example: Heart dataset (iii)

- Best tree: six nodes



Categorical predictors

- q values: $2^q - 1$ possible partitions into two groups
- In a two-class problem: order the predictor classes according to the proportion falling in outcome class 1
 - Then split as an ordered predictor
 - Optimal split in terms of Gini index and cross-entropy
- This also holds for regression (RSS)
 - Order the categories by increasing mean of the outcome
- For multi-category outcomes no such simplifications are possible
- Try to avoid variables with large q : overfitting

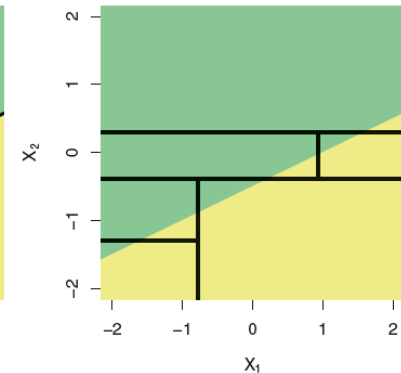
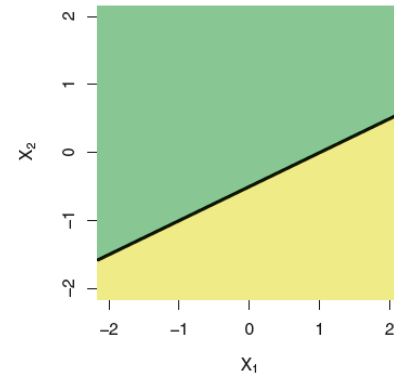
Trees vs. Linear Models

- Linear regression:
$$f(X) = \beta_0 + \sum_{j=1}^p X_j \beta_j$$

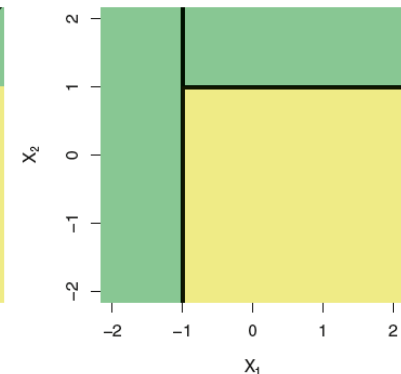
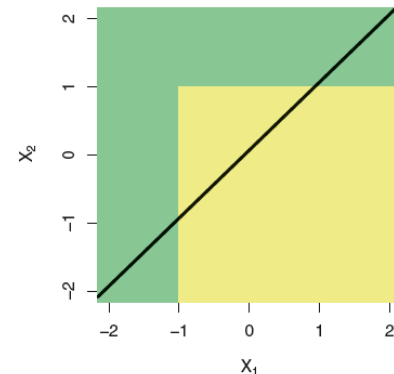
- Regression trees:
$$f(X) = \sum_{m=1}^M c_m \cdot 1_{(X \in R_m)}$$

- A two-dimensional classification example:

Linear model is better



Tree is better



Advantages and Disadvantages of trees

■ Pros:

- Easy to explain
- More closely mirror human decision-making
- Can be displayed graphically, and are easily interpreted
- Can handle qualitative predictors without the need to create dummy variables

■ Cons:

- Predictive accuracy is lower than other approaches

Bibliography

- G. James, D. Witten, T. Hastie, y R. Tibshirani, An Introduction to Statistical Learning with Applications in R. Springer, 2013.
 - Chapter 8, Sec. 8.1.
- T. Hastie, R. Tibshirani, y J. Friedman, The elements of statistical learning. Springer, 2009.
 - Chapter 9, Sec. 9.2.