

Lecture 15. Decision Trees¶

CS 109A/AC 209A/STAT 121A Data Science:

Harvard University

Fall 2016

Instructors: P. Protopapas, K. Rader, W. Pan

Announcements

- Projects. Milestone 3 is due Saturday
- HW5 solutions coming soon. Few students had medical emergencies so we will not release until everyone is done

Quiz

Code: willbyers

Outline

Tree based methods

Regression Trees

Classification Trees

Bagging

Outline

Tree based methods

Regression Trees

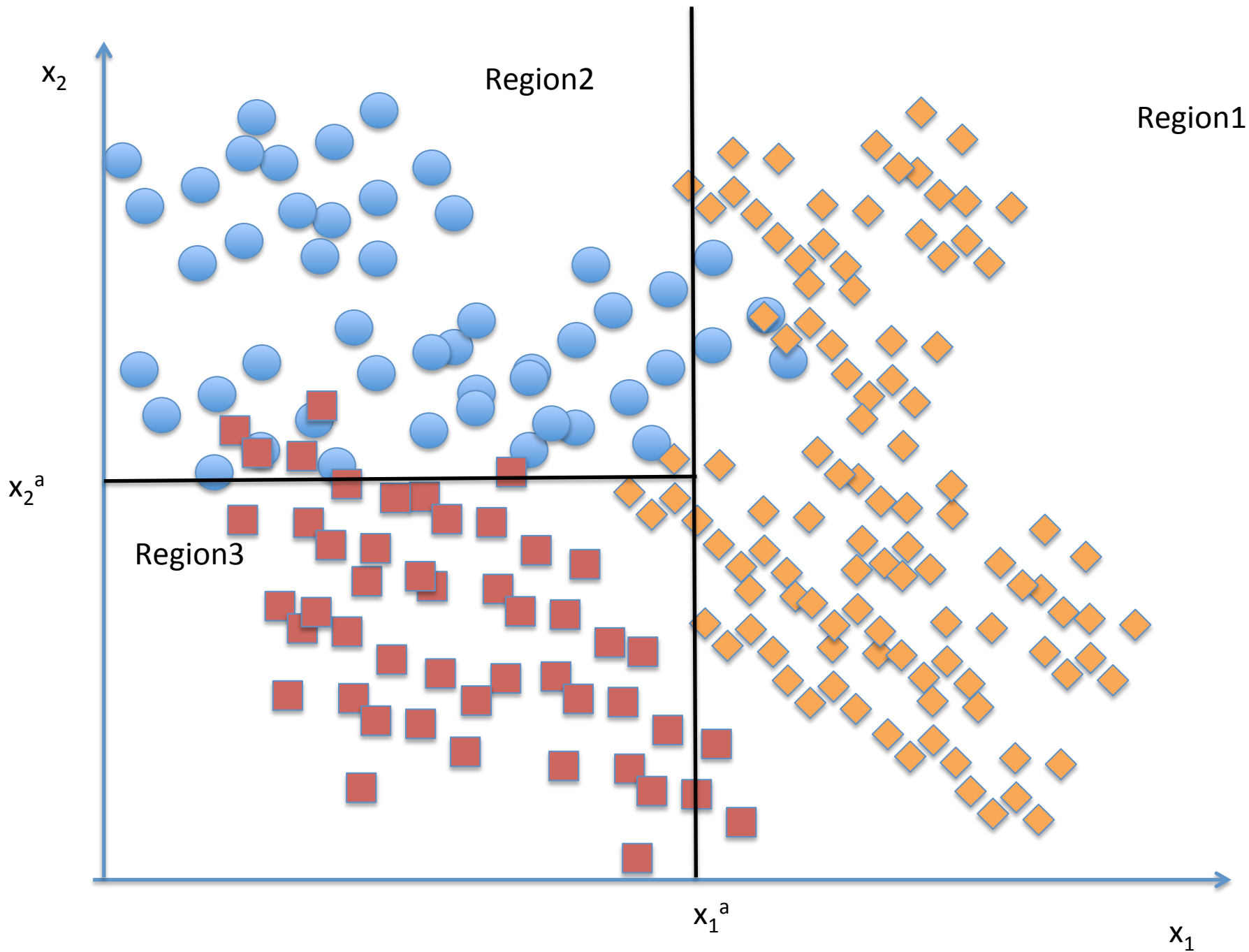
Classification Trees

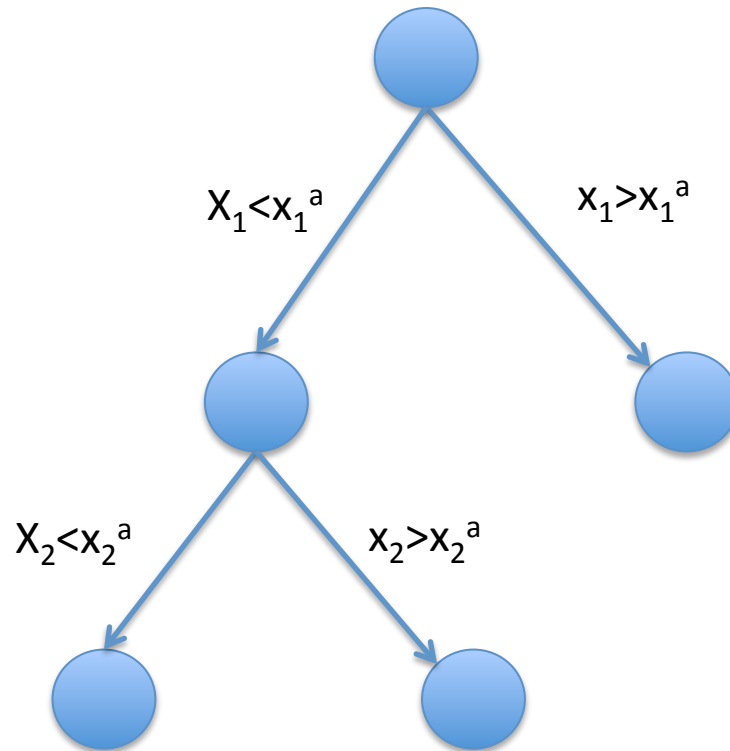
Bagging

Tree-based methods



Basic Idea





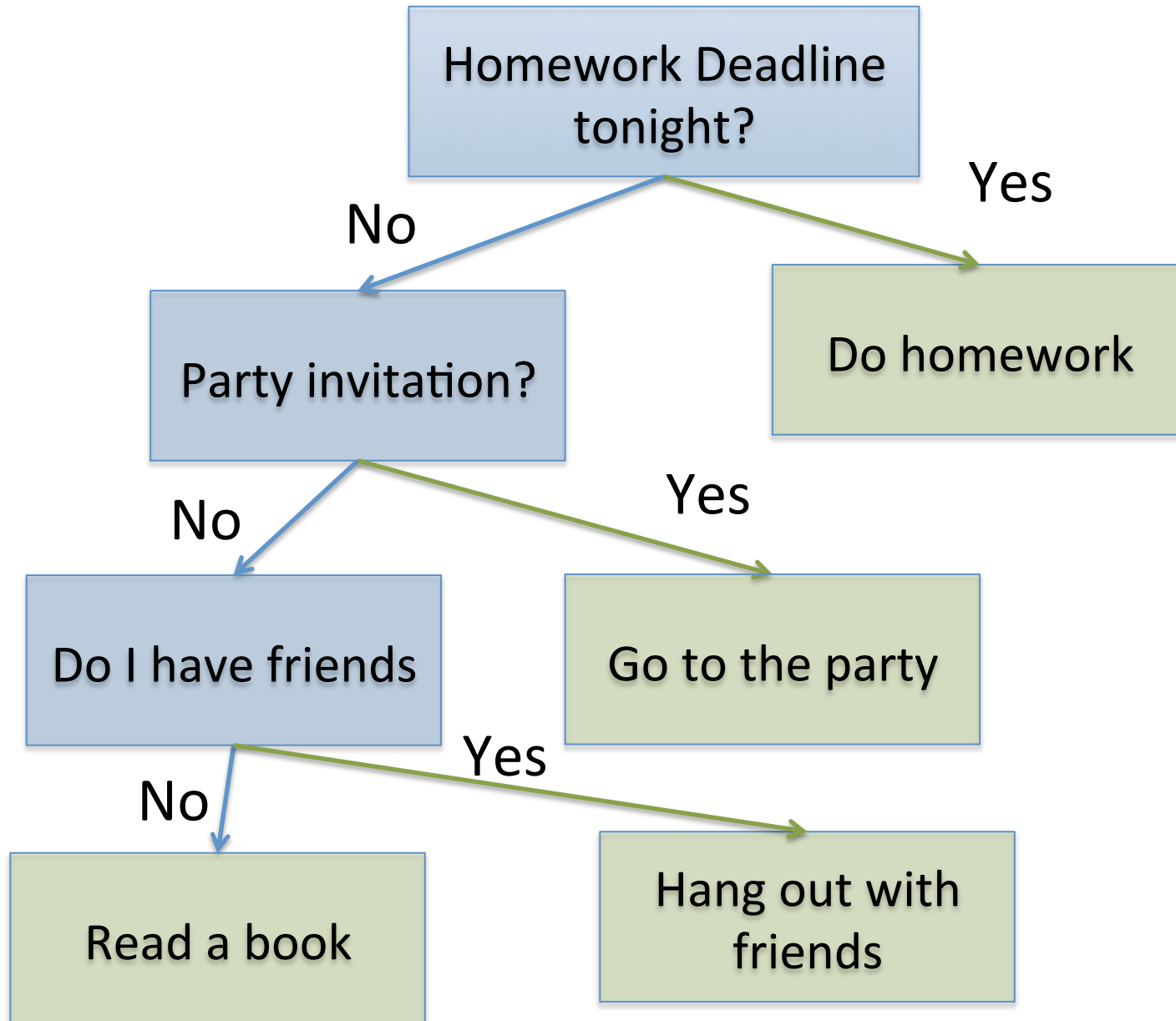
Basic Idea

Segment the predictor space into sub-regions and we learn from the training set the value to predict as the mean or mode or median of the respond variable of the training examples that are in that segment.

Why Trees?

What would you do tonight? Decide amongst the following:

- Finish homework
- Go to a party
- Read a book
- Hang out with friends



Why Trees?

We split the predictor space as branches of a tree and therefore these methods are called decision tree methods

Why Forest?

Not the most powerful models but using multiple trees as in bagging, random forests and boosting yield much better results.



Outline

Tree based methods

Regression Trees

Classification Trees

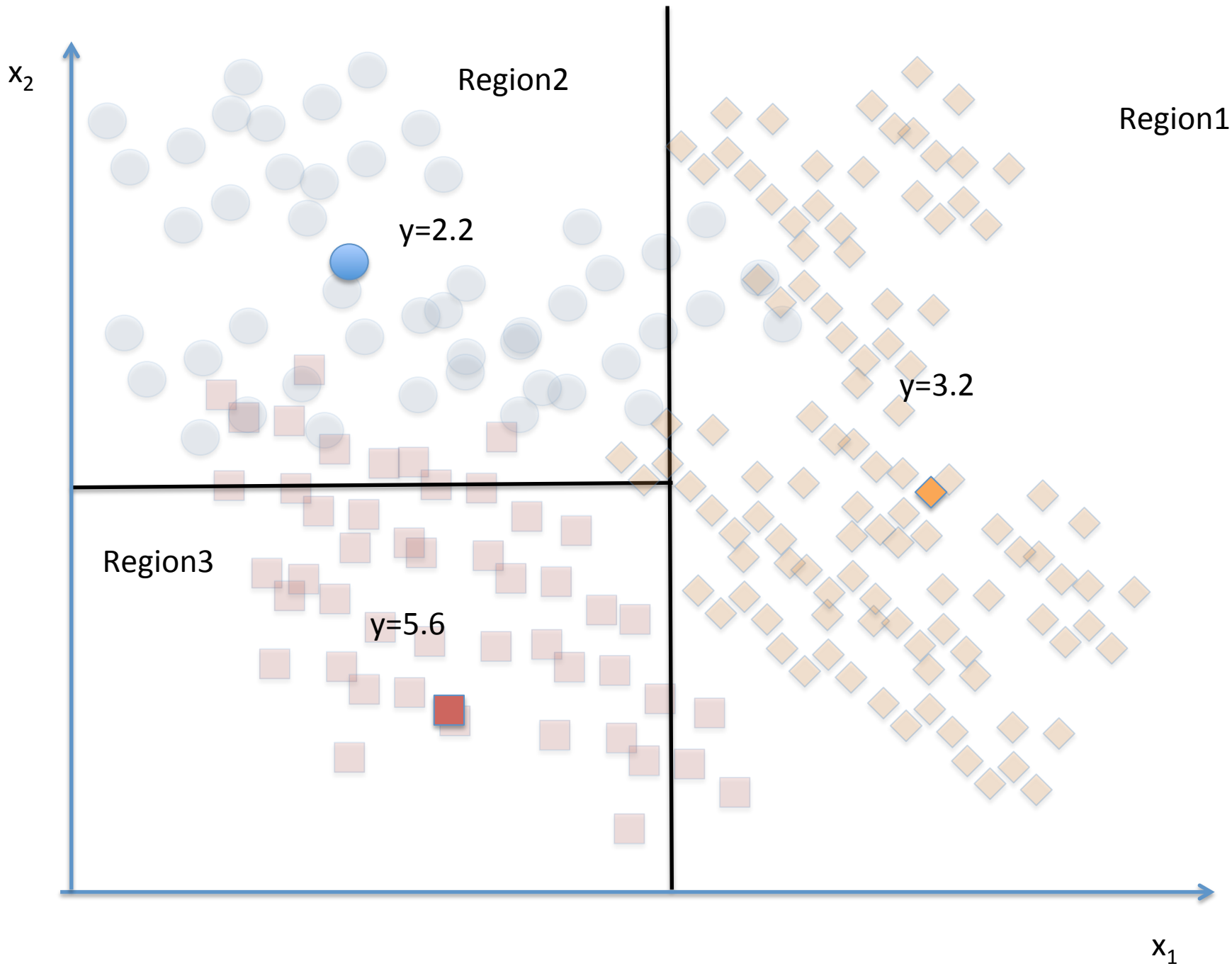
Bagging

Regression

Build a regression tree:

Divide the predictor space into J distinct not overlapping regions $R_1, R_2, R_3, \dots, R_J$

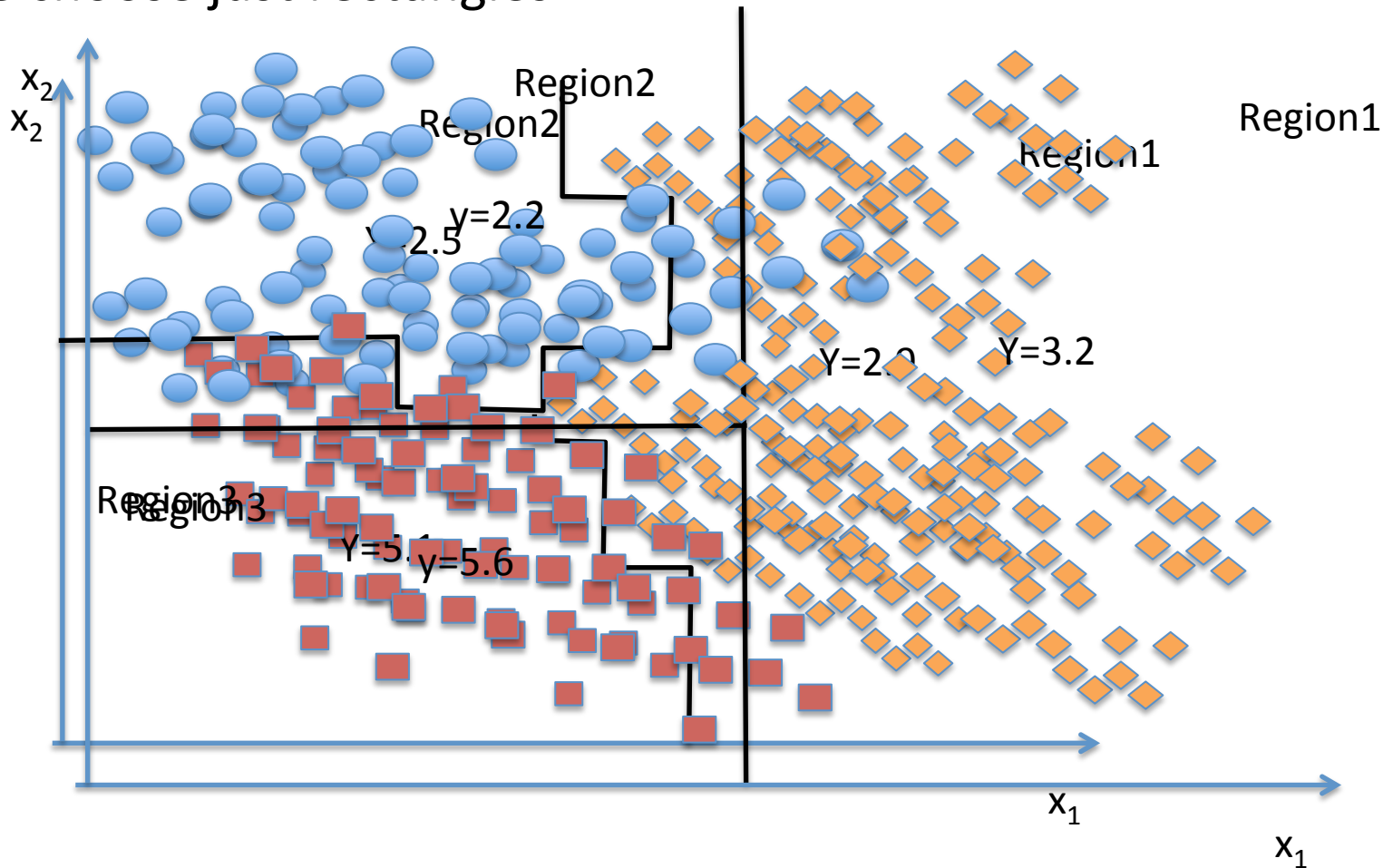
We make the same prediction for all observations in the same region; use the mean of responses for all training observations that are in the region



Finding the sub-regions

The regions could have any shape.

But we choose just rectangles



Find boxes R_1, \dots, R_J that minimize the RSS

$$RSS = \sum_{j=1}^J \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

where \hat{y}_{R_j} is the mean response value of all training observations in the R_j region

This computationally very expensive!

Solution: Top down approach, greedy approach
recursive binary splitting

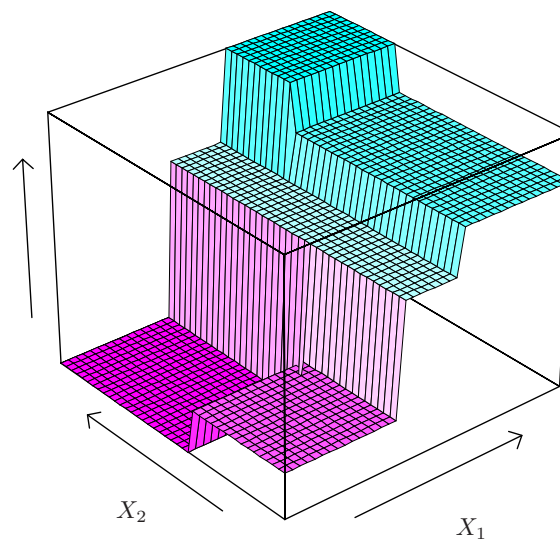
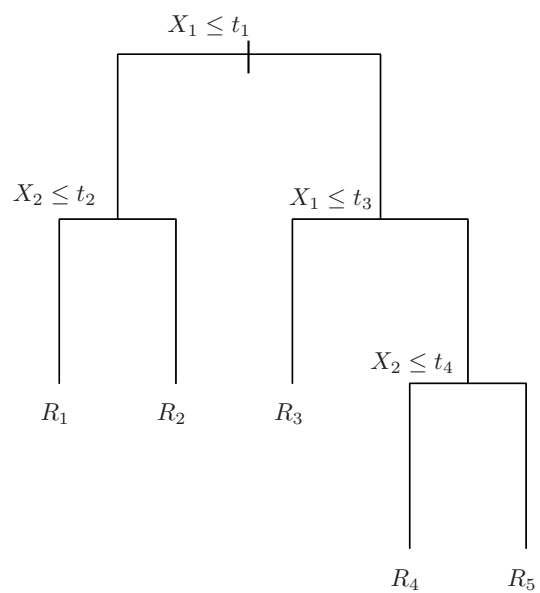
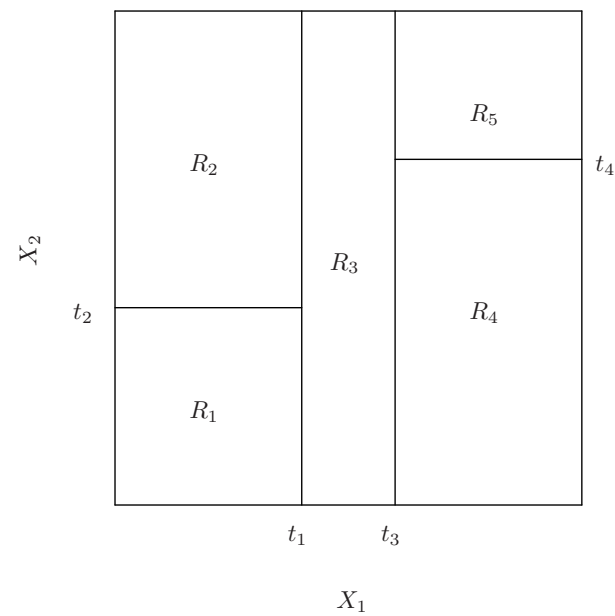
Recursive Binary Splitting

1. Consider all predictor X_p and all the all possible values of the cutpoints s for each of the predictors. Choose the predictor and cutpoint s.t. it minimizes the RSS

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

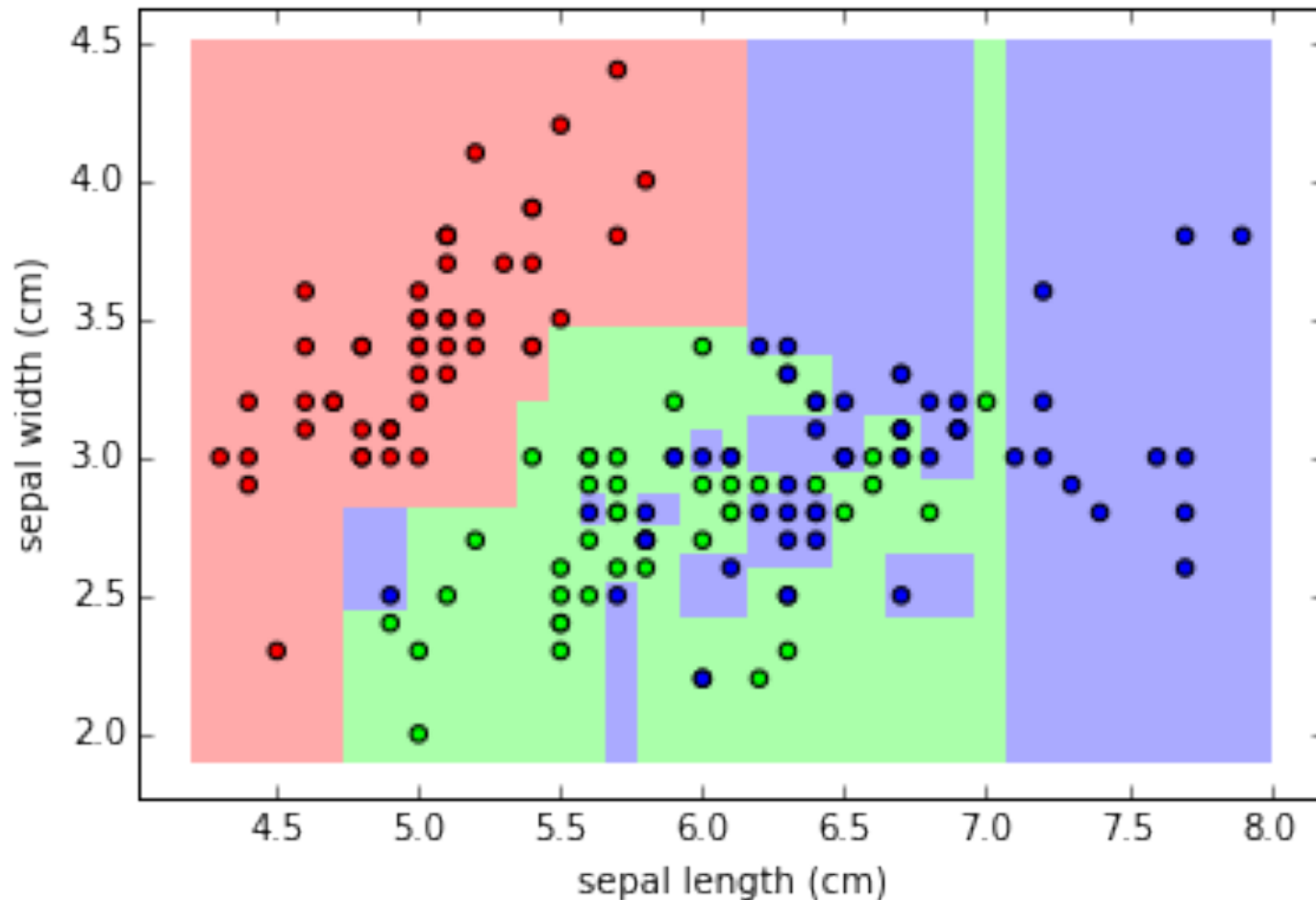
This can be done quickly, assuming number of predictors is not very large

2. Repeat #1 but only consider the sub-regions
3. Stop: node contains only one class or node contains less than n data points or max depth is reached



Overfitting

If we keep splitting we will be reducing RSS



Pruning

Fewer splits or fewer regions lower variance better interpretation at cost of little more bias

Ideas?

Stop splitting when RSS improvement is lower than a threshold

- Smaller trees but not effective (short sighted)

- A split early on in the tree might be followed by a very good split; a split that leads to a large reduction in RSS later on

Pruning

Better is to grow a large tree and then look subtrees that minimize the **test error**

How?

Cross-validation of all possible subtrees?

This is too expensive

Cost complexity pruning—also known as weakest link pruning

Cost complexity pruning

Consider a tuning parameter α that for each value of α there is a subtree that minimizes

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

Where $|T|$ is the number of terminal nodes. α controls the complexity of the tree similarly we saw with other regularizations (e.g. LASSO).

It turns out that as we increase α from zero in, branches get pruned from the tree in a nested and predictable fashion, so obtaining the whole sequence of subtrees as a function of α is easy.

ALGORITHM FOR PRUNING

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α
3. Use K-fold cross-validation to choose α
 - Repeat #1 and #2 on the k-th fold
 - Estimate the MSE as a function of αAverage all and pick α
4. Return the subtree from Step 2 that corresponds to the chosen value of α

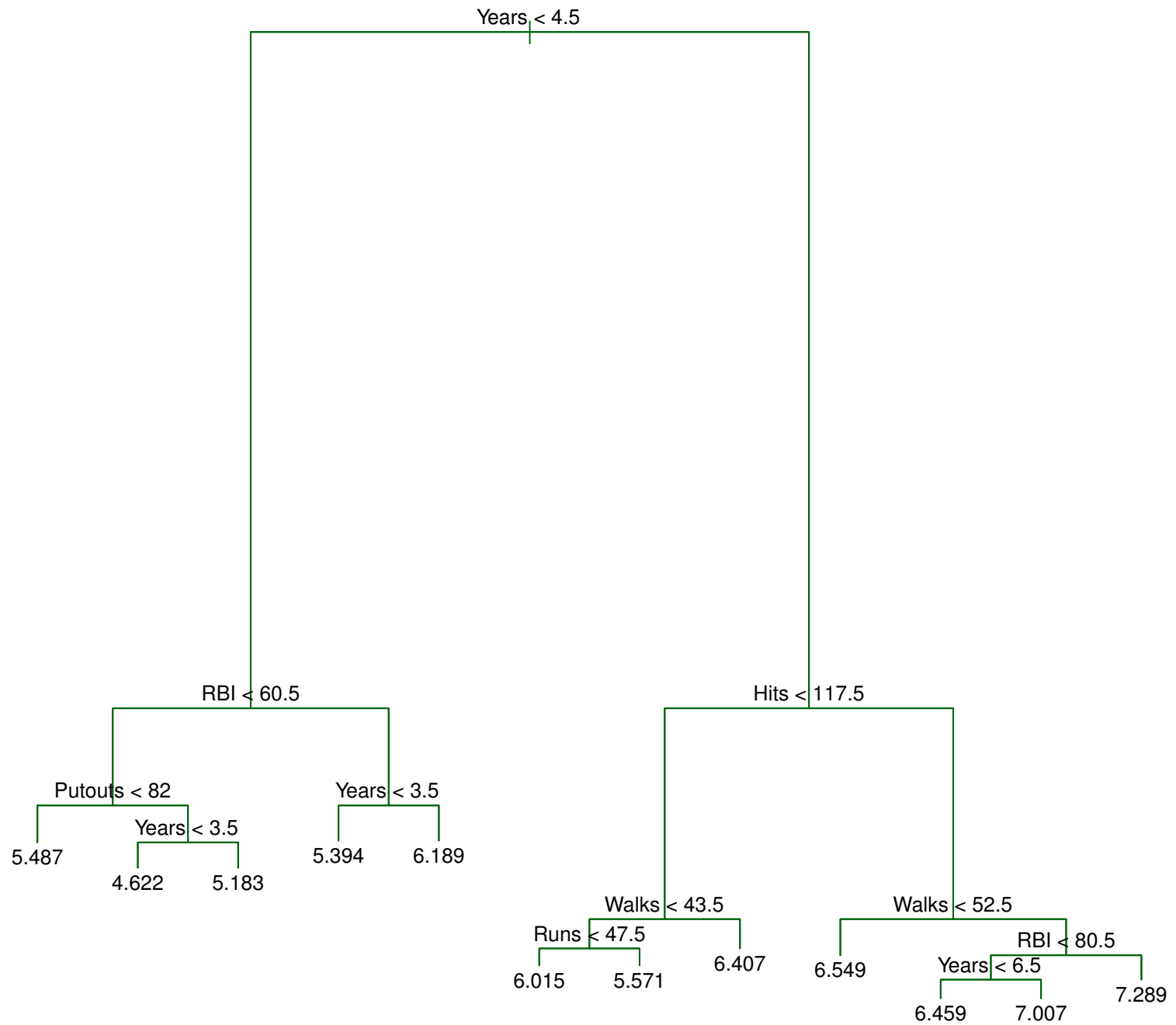
Hitters data set:

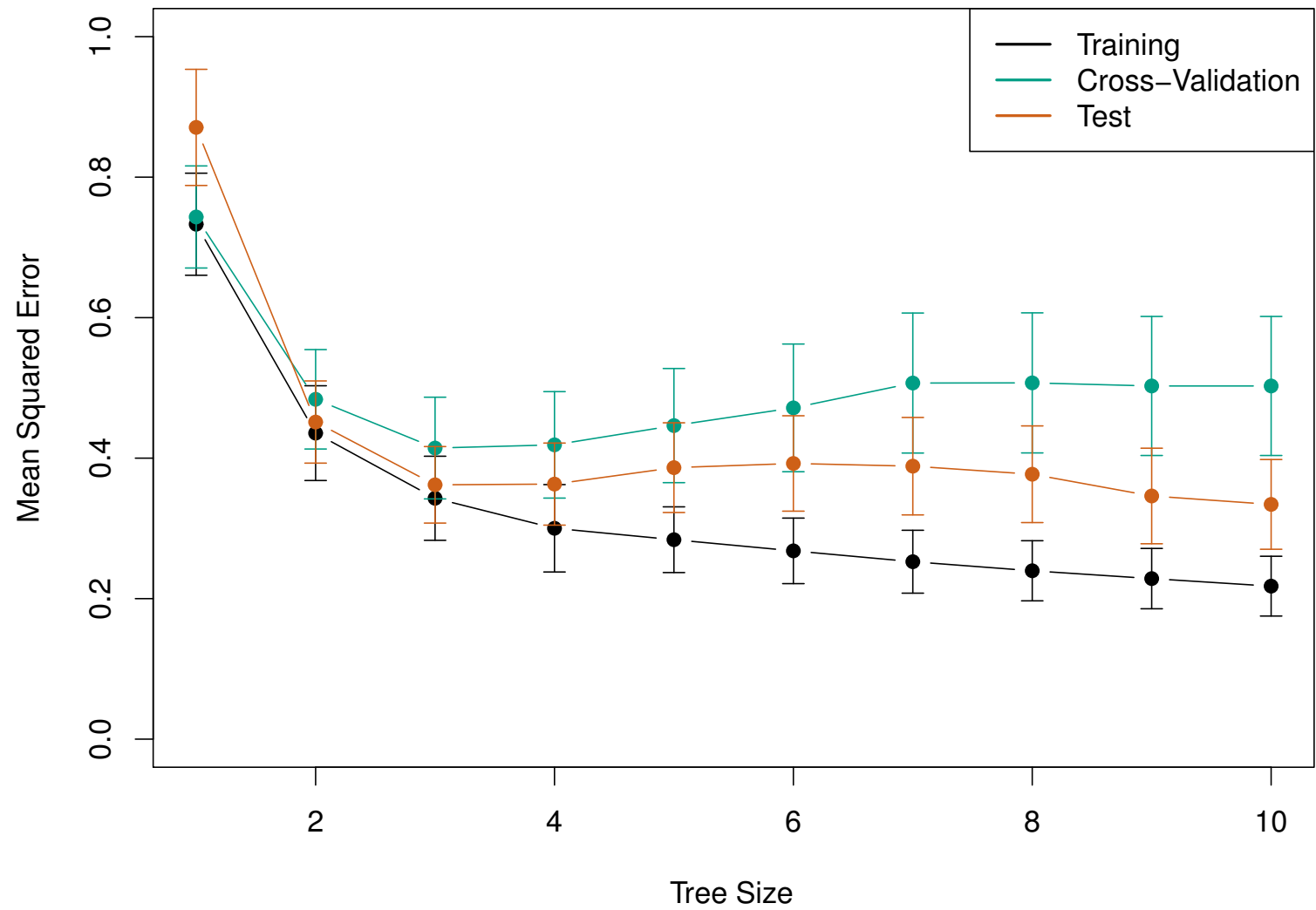
Response variable: baseball player's Salary

Predictors:

- Years (the number of years that he has played in the major leagues)
- Hits (the number of hits that he made in the previous year)
- Walks, RBI, hits, putouts

Note: log-transform Salary so that its distribution has more of a typical bell-shape.





Outline

Tree based methods

Regression Trees

Classification Trees

Bagging

Classification Trees

Very similar to regression except that it is used to predict a **qualitative** response rather than a quantitative one

In regression trees we use the *mean response* of the training observations for classification trees we use **most commonly occurring class**

Interested in the class proportions of each region

Classification Trees

We learn the model using recursive binary splitting as with the regression trees except ...

RSS cannot be used as a criterion for making the binary splits.

Classification error rate:

$$E = 1 - \max_k \hat{p}_{mk}$$

\hat{p}_{mk} represents the proportion of training observations in the m -th region that are from the k -th class

Gini index

Classification error is not differentiable or sensitive enough for tree growing.

Purity of the nodes, Gini index

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

G takes small values when p_{mk} is small or close to 1, therefore is a measure of purity of the nodes.

Gini index

Example:

5 red, 2 blue and 3 green

red:


$$5/10 * (1 - 5/10) = 0.25$$

blue:

$$2/10 * (1 - 2/10) = 0.16$$

green:

$$3/10 * (1 - 3/10) = 0.21$$


$$G = 0.61$$

Misclassification

Example:

5 red, 2 blue and 3 green

Proportions p_{mk} for each class are: 5/10, 2/10, 3/10

$$\max_k(p_{mk}) = 5/10$$

Classification error

$$E = 1 - 1/2 = 1/2$$

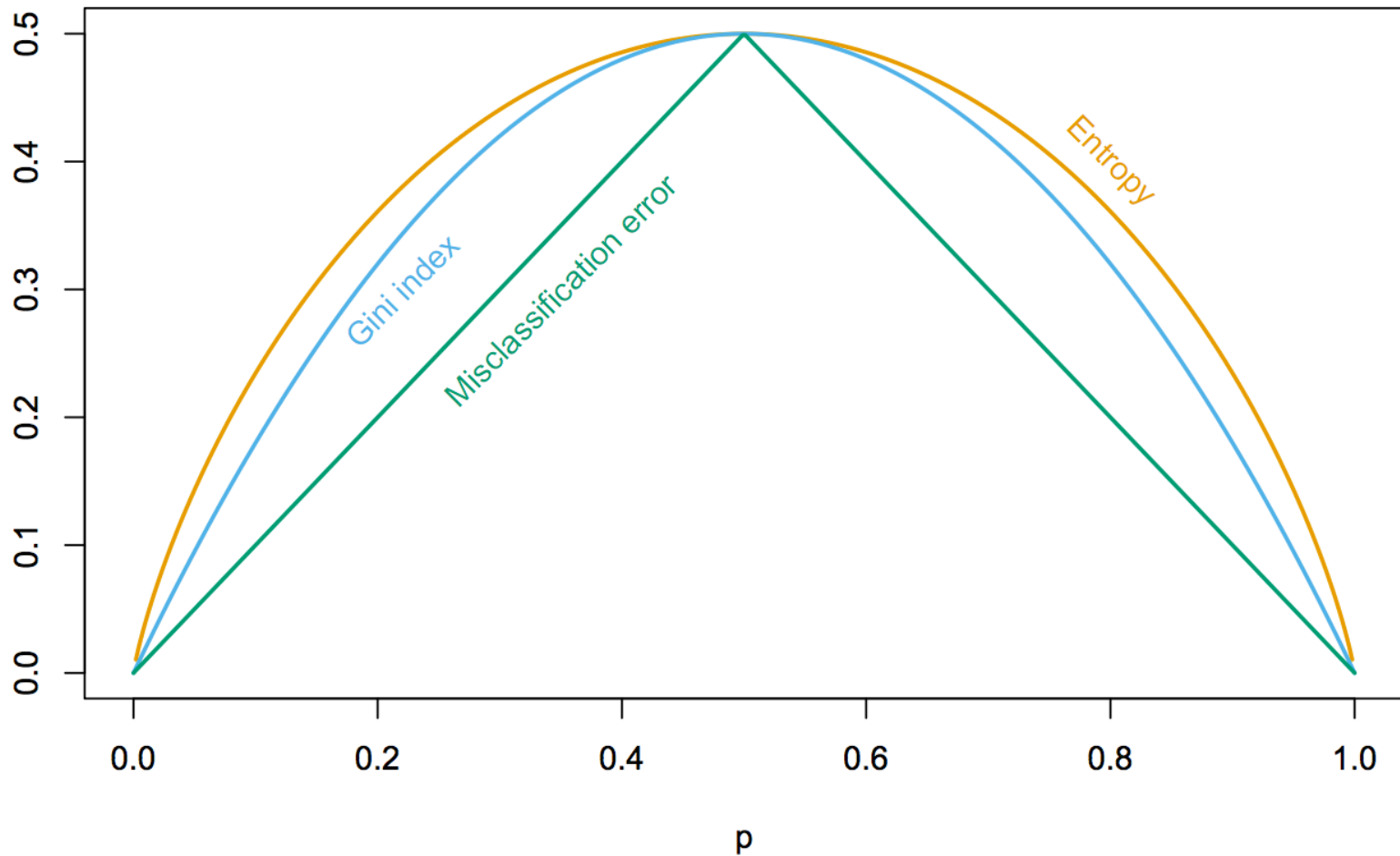
Cross entropy

Alternative to the Gini index is cross entropy

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

$D > 0$ and will take value near zero when p_{mk} is either near zero or one

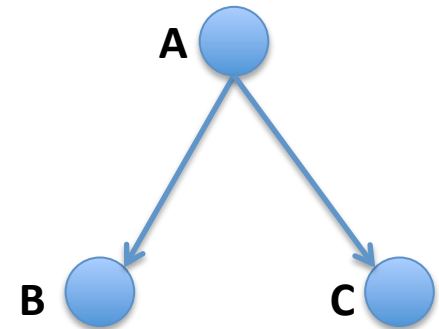
Node impurity for two class problem



Node Purity Gain

Compare:

- Gini impurity of parent node
- Gini impurity of child nodes



$$\Delta G = G(A) - \frac{N_B}{N_A} G(B) - \frac{N_C}{N_A} G(C)$$

Pruning classification Tree

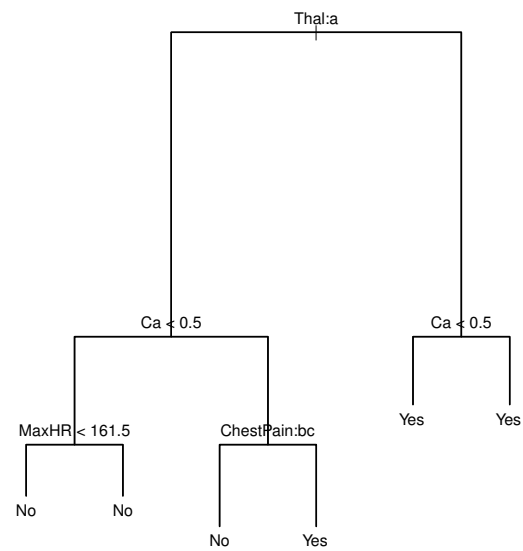
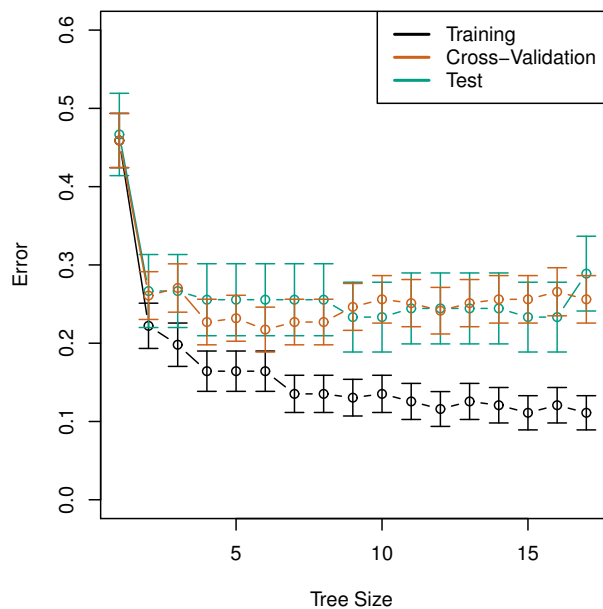
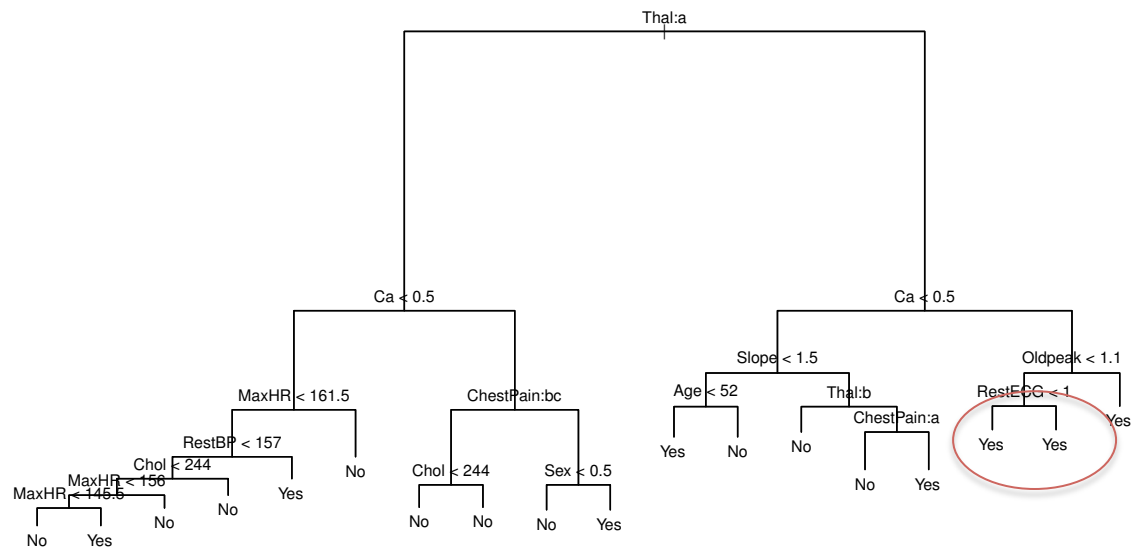
- Use the same algorithm as for regression tree but instead of RSS use Gini index or Entropy

HOWEVER: classification error rate is preferable for the final pruned tree

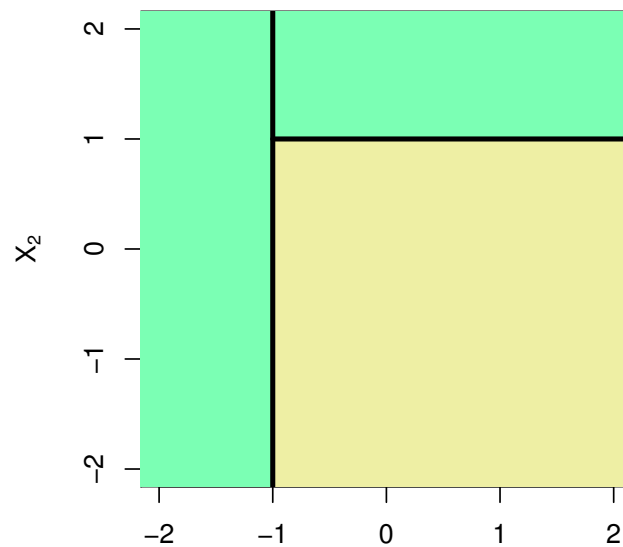
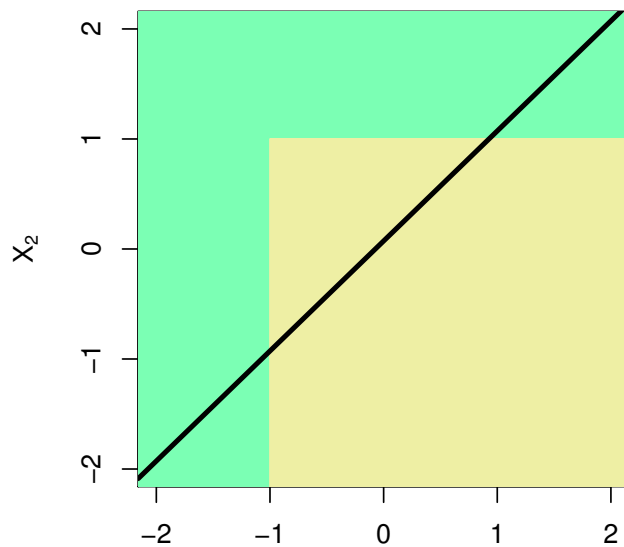
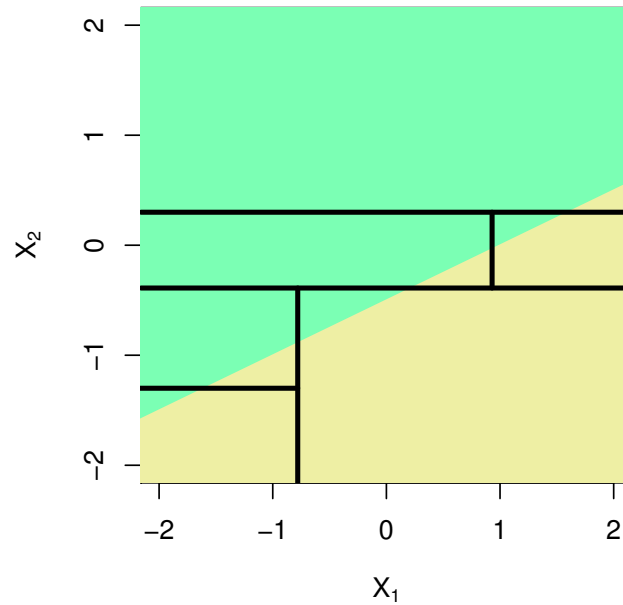
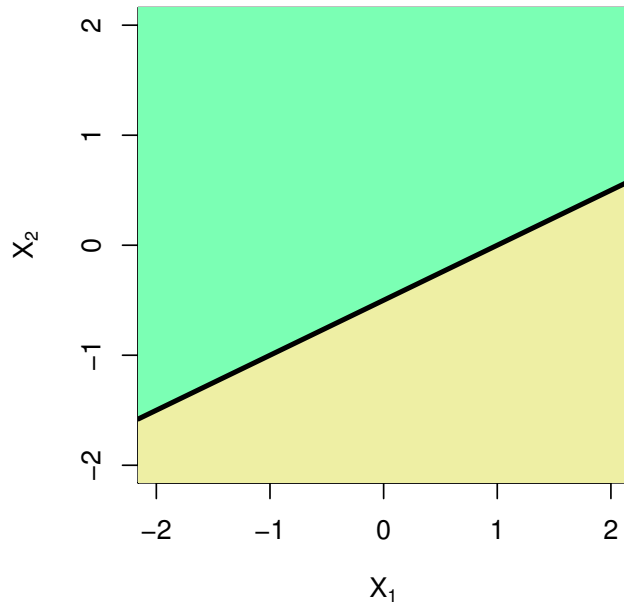
Example: *Heart data set*

- 303 patients who presented with chest pain
- Response takes values Yes/No indicates the presence of heart disease
- 13 predictors including Age, Sex, Chol (a cholesterol measurement), and other heart and lung function measurements

Cross validation



Comparison to linear models



Outline

Tree based methods

Regression Trees

Classification Trees

Bagging

Advantages of tree based methods

- Easy to explain
- Handle qualitative predictors
- Display graphically and easy to interpret
- More similar to human decision making

Disadvantages

- Non robust. Sensitive to small changes in the data
- Trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches we have seen
- Only axis aligned splits

Power of the crowds



Ensemble methods

- A single decision tree does not perform well
- But, it is super fast
- What if we learn multiple trees?

We need to make sure they do not all just learn the same

Bagging

If we split the data in random different ways, decision trees give different results, **high variance**.

Bagging: Bootstrap **aggregating** is a method that result in low variance.

If we had multiple realizations of the data (or multiple samples) we could calculate the predictions multiple times and take the average of the fact that averaging multiple onerous estimations produce less uncertain results

Bagging

Say for each sample b , we calculate $f^b(x)$, then:

$$\hat{f}_{avg}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x)$$

How?

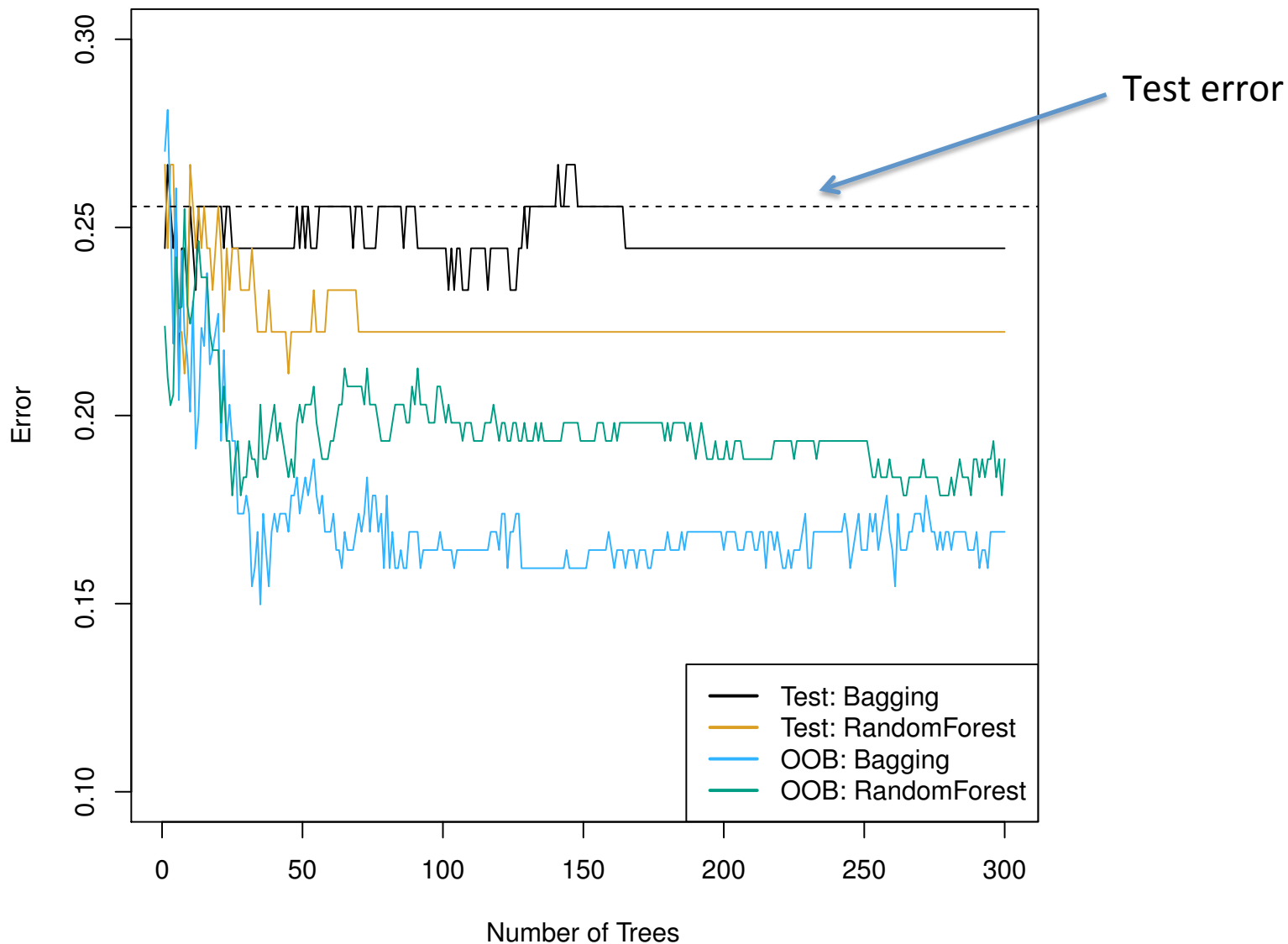
Bootstrap

Construct B (hundreds) of trees (no pruning)

Learn a classifier for each bootstrap sample and average them

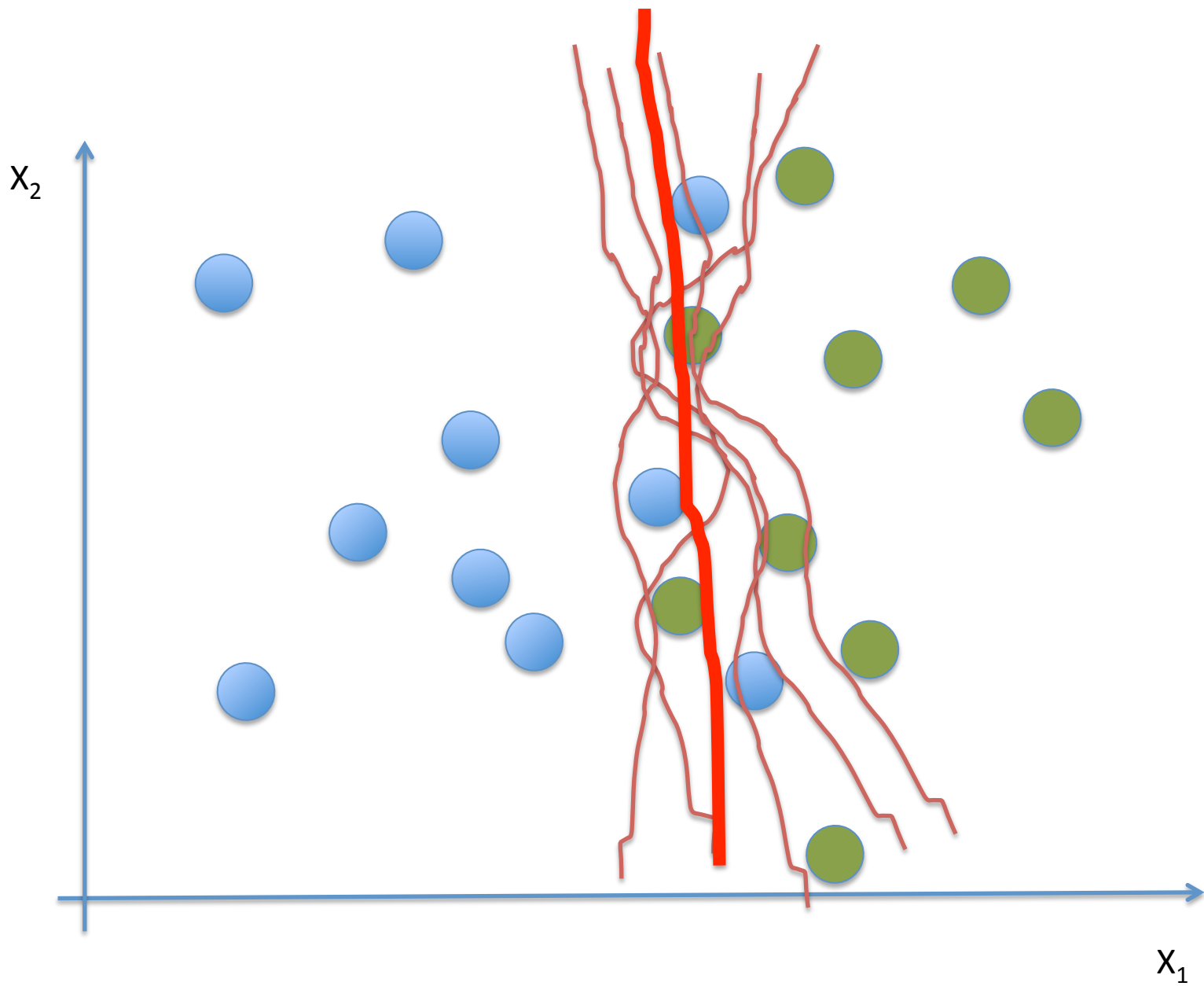
Very effective

Bagging for classification: Majority vote

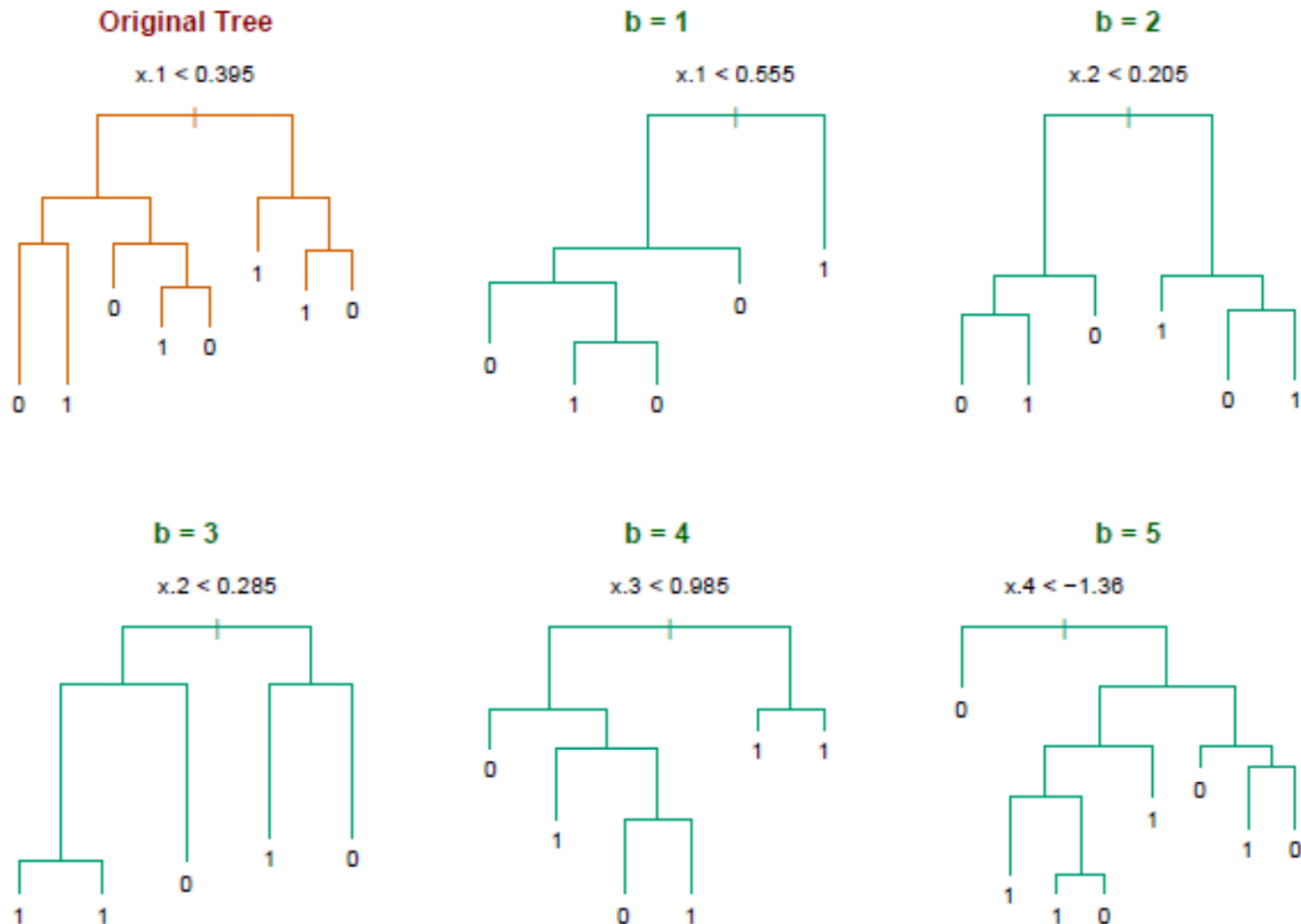


Out-of-Bag Error Estimation

- No cross validation?
- Remember, in bootstrapping we sample with replacement, and therefore **not all observations are used for each bootstrap sample**. On average $1/3$ of them are not used!
- We call them out-of-bag samples (OOB)
- We can predict the response for the i -th observation using each of the trees in which that observation was OOB and do this for n observations
- Calculate overall OOB MSE or classification error



Bagging decision trees



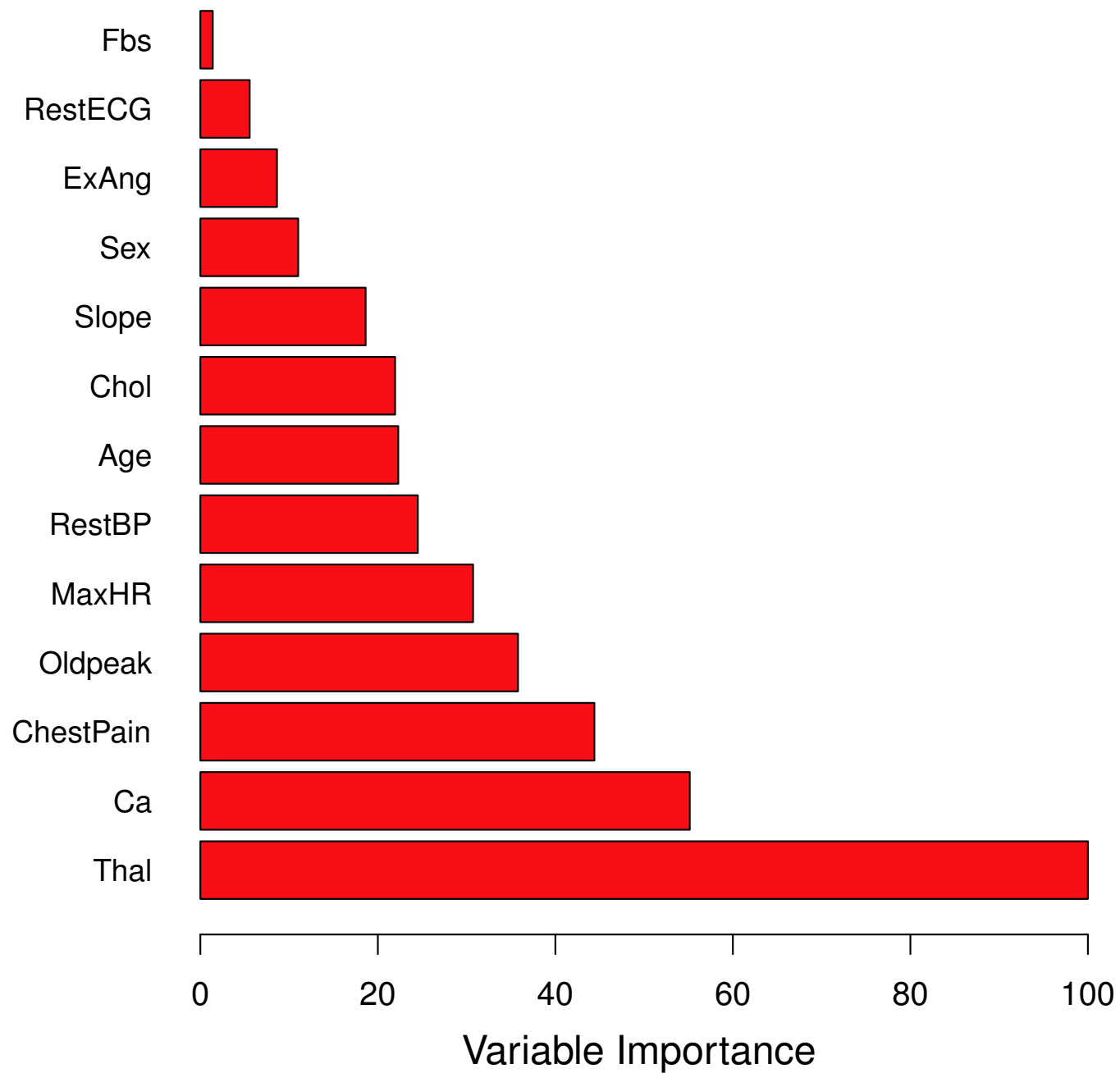
Bagging

- Reduces overfitting (variance)
- Normally uses one type of classifier
- Decision trees are popular
- Easy to parallelize

Variable Importance Measures

- Bagging results in improved accuracy over prediction using a single tree
- Unfortunately, difficult to interpret the resulting model. Bagging improves prediction accuracy at the expense of interpretability.

Calculate the total amount that the RSS or Gini index is decreased due to splits over a given predictor, averaged over all B trees.



Different flavors

- ID3, or alternative Dichotomizer, was the first of three Decision Tree implementations developed by Ross Quinlan (Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.)
- C4.5, Quinlan's next iteration. The new features (versus ID3) are: (i) accepts both continuous and discrete features; (ii) handles incomplete data points; (iii) solves over-fitting problem by (very clever) bottom-up technique usually known as "pruning"; and (iv) different weights can be applied the features that comprise the training data.

Used in orange <http://orange.biolab.si/>

- CART or Classification And Regression Trees is often used as a generic acronym for the term Decision Tree, though it apparently has a more specific meaning. In sum, the CART implementation is very similar to C4.5; the one notable difference is that CART uses the Gini index.

Used in sklearn

Further reading

- Pattern Recognition and Machine Learning,
Christopher M. Bishop
- The Elements of Statistical Learning
Trevor Hastie, Robert Tibshirani, Jerome Friedman
[http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/
ESLII_print10.pdf](http://statweb.stanford.edu/~tibs/ElemStatLearn/printings/ESLII_print10.pdf)