

# Ch 8.2.1, 8.2.2: Bagging and Random Forests

## Lecture 24 - CMSE 381

Prof. Elizabeth Munch

Michigan State University

::

Dept of Computational Mathematics, Science & Engineering

Mon, Nov 14, 2022

## Last time:

- 8.1 Decision Trees

## This lecture:

- 8.2.1 Bagging
- 8.2.2 Random forest

## Announcements:

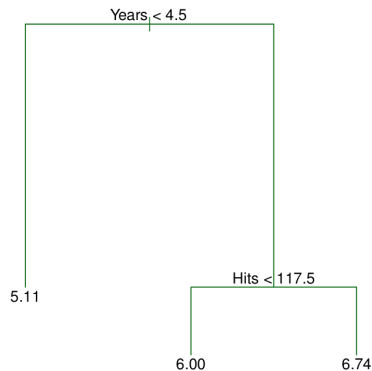
Status	Lec #	Date		Reading	Homeworks
		Mon Oct 23	No class - Fall break		
		Wed Oct 25	<b>Midterm #2</b>		
Done	20	Fri Oct 27	Dimension Reduction	6.3	
Done	21	Mon Oct 30	More dimension reduction; High dimensions	6.4	
Done	22	Wed Nov 1	Polynomial & Step Functions	7.1, 7.2	
Pushed	23	Fri Nov 3	Step Functions; Basis functions; Start Splines	7.2 - 7.4	
	24	Mon Nov 6	Regression Splines	7.4	HW #6-Due
	25	Wed Nov 8	Decision Trees	8.1	HW #6 Due
	26	Fri Nov 10	Random Forests	8.2.1, 8.2.2	
	27	Mon Nov 13	Maximal Margin Classifier	9.1	
	28	Wed Nov 15	SVC	9.2	
	29	Fri Nov 17	SVM	9.3, 9.4	
	30	Mon Nov 20	Single layer NN	10.1	
	31	Wed Nov 22	Virtual: Project office hours		
		Fri Nov 24	No class - Thanksgiving		
		Mon Nov 27	<b>Review</b>		
		Wed Nov 29	<b>Midterm #3</b>		
	32	Fri Dec 1	Multi Layer NN	10.2	
	33	Mon Dec 4	CNN	10.3	
	34	Wed Dec 6	Unsupervised Learning & Clustering	12.1, 12.4	
	35	Fri Dec 8	Virtual: Project office hours		Project due

# Section 1

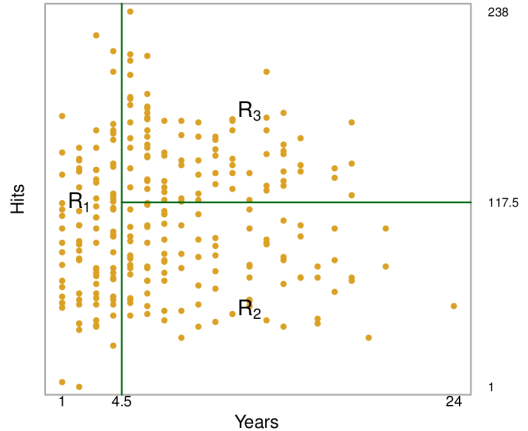
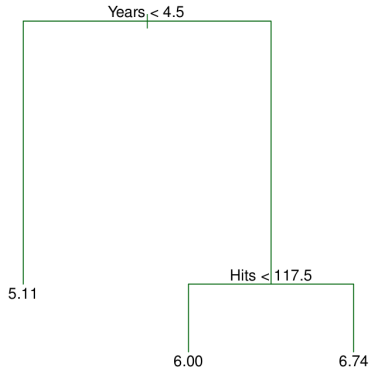
Last time

# First decision tree example

	Hits	Years	LogSalary
1	81	14	6.163315
2	130	3	6.173786
3	141	11	6.214608
4	87	2	4.516339
5	169	11	6.620073
...	...	...	...
317	127	5	6.551080
318	136	12	6.774224
319	126	6	5.953243
320	144	8	6.866933
321	170	11	6.907755

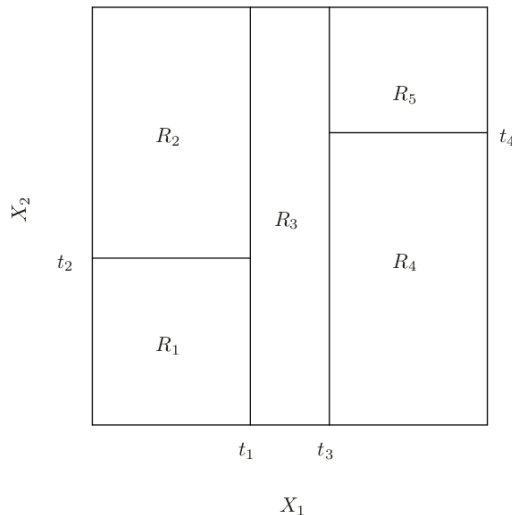


# Viewing Regions Defined by Tree



# How do we actually get the tree? Two steps

- 1 We divide the predictor space — that is, the set of possible values for  $X_1, X_2, \dots, X_p$  — into  $J$  distinct and non-overlapping regions,  $R_1, R_2, \dots, R_J$ .
- 2 For every observation that falls into the region  $R_j$ , we make the same prediction = the mean of the response values for the training observations in  $R_j$ .



# Recursive binary splitting

## Goal:

Find boxes  $R_1, \dots, R_J$  that minimize

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

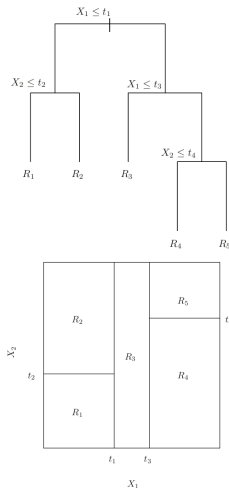
$\hat{y}_{R_j}$  = mean response for training observations in  $j$ th box

Pick  $s$  so that splitting into  $\{X \mid X_j < s\}$  and  $\{X \mid X_j \geq s\}$  results in largest possible reduction in RSS:

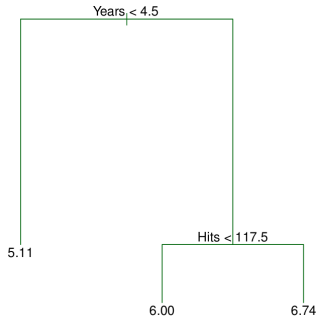
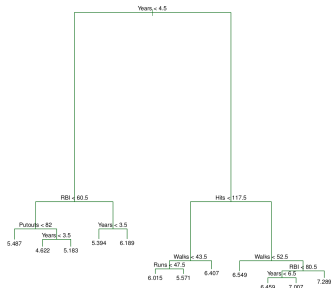
$$R_1(j, s) = \{X \mid X_j < s\}$$

$$R_2(j, s) = \{X \mid X_j \geq s\}$$

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

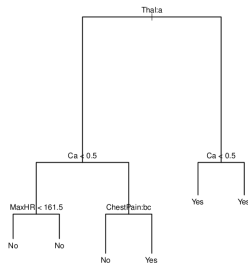
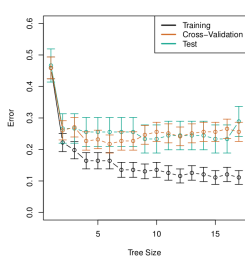
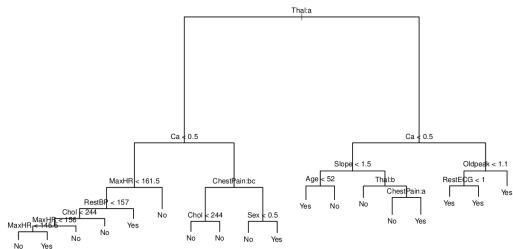


# Pruning





# Classification version

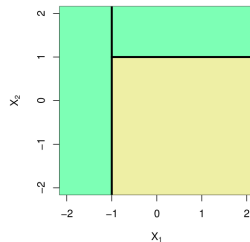
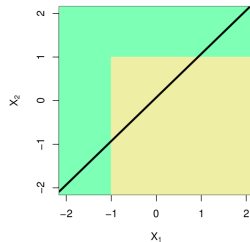
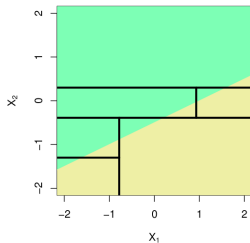
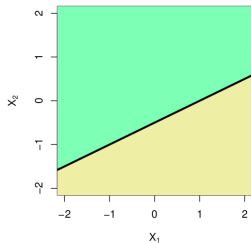


## Evaluating the splits:

- $\hat{p}_{mk}$  = proportion of training observations in  $R_m$  from the  $k$ th class
- Error:  $E = 1 - \max_k(\hat{p}_{mk})$
- Gini index:

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

# Linear models vs trees



**Pros:**

**Cons:**

## Section 2

### 8.2.1 Bagging

# Recall: The bootstrap

## Want to do (but can't):

Build separate models from independent training sets, and average resulting predictions:

- $\hat{f}^1(x), \dots, \hat{f}^B(x)$  for  $B$  separate training sets
- Return the average

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

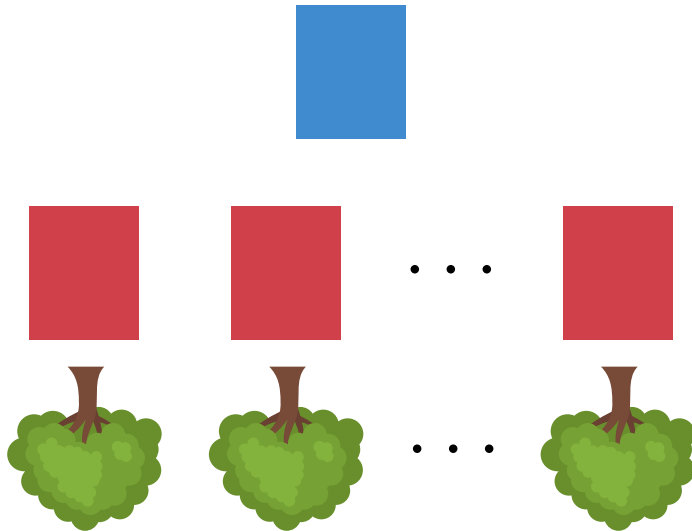
## Bootstrap modification:

- Work with fixed data set
- Take  $B$  samples from this data set (with replacement)
- Train method on  $b$ th sample to get  $\hat{f}^{*b}(x)$
- Return average of predictions (regression)

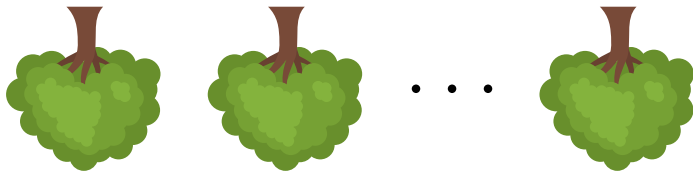
$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$$

or majority vote (classification)

# Tree version



# Prediction on new data point



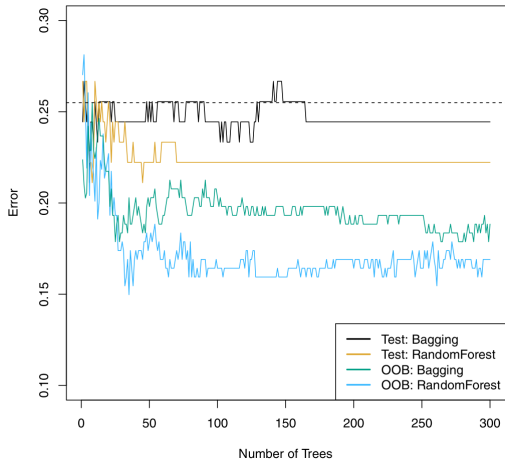
# Bagging vs Bootstrap

**Bootstrap**

**Bagging**

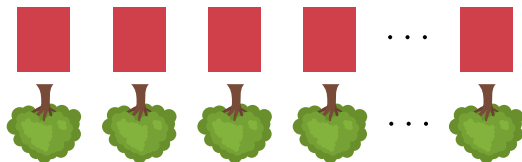


## Example: Heart classification data

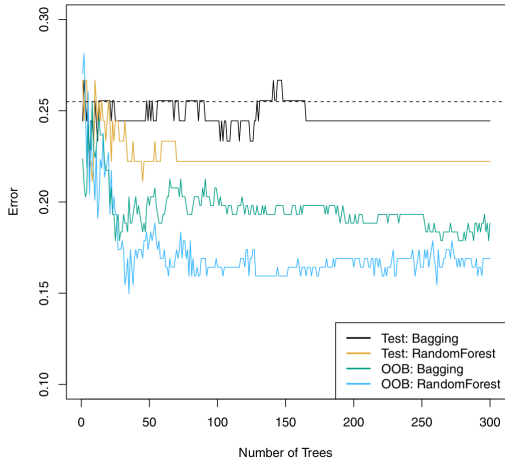


# Out of Bag Error Estimation

- On average, bootstrap sample uses about  $2/3$  of the data
- Remaining observations not used are called *out-of-bag* (OOB) observations
- For each observation, run through all the trees where it wasn't used for building
- Return the average (or majority vote) of those as test prediction



# Error using OOB



# Bagging code example

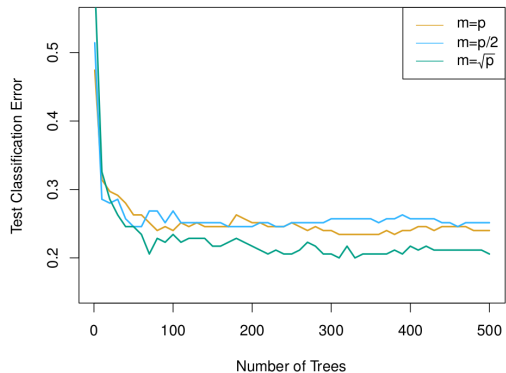
## Section 3

### Random Forests

# The idea

- Goal is to decorrelate the bagged trees:
  - ▶ If there is a strong predictor, the first split of most trees will be the same
  - ▶ Most or all trees will be highly correlated
  - ▶ Averaging highly correlated quantities doesn't decrease variance as much as uncorrelated
- The random forrest fix:
  - ▶ Each time a split is considered, only use a random subset of  $m$  the predictors
  - ▶ Fresh sample taken every time
  - ▶ Typically  $m \approx \sqrt{p}$
  - ▶ On average,  $(p - m)/p$  of splits won't consider strong predictor
  - ▶  $m = p$  gives back bagging

# Example on gene expression



# Coding example for random forests





# Next time

Status	Lec #	Date		Reading	Homeworks
		Mon	Oct 23	No class - Fall break	
		Wed	Oct 25	<b>Midterm #2</b>	
Done	20	Fri	Oct 27	Dimension Reduction	6.3
Done	21	Mon	Oct 30	More dimension reduction; High dimensions	6.4
Done	22	Wed	Nov 1	Polynomial & Step Functions	7.1, 7.2
Pushed	23	Fri	Nov 3	Step Functions; Basis functions; Start Splines	7.2 - 7.4
	24	Mon	Nov 6	Regression Splines	7.4
	25	Wed	Nov 8	Decision Trees	8.1
	26	Fri	Nov 10	Random Forests	8.2.1, 8.2.2
	27	Mon	Nov 13	Maximal Margin Classifier	9.1
	28	Wed	Nov 15	SVC	9.2
	29	Fri	Nov 17	SVM	9.3, 9.4
	30	Mon	Nov 20	Single layer NN	10.1
	31	Wed	Nov 22	Virtual: Project office hours	
		Fri	Nov 24	No class - Thanksgiving	
		Mon	Nov 27	<b>Review</b>	
		Wed	Nov 29	<b>Midterm #3</b>	
	32	Fri	Dec 1	Multi Layer NN	10.2
	33	Mon	Dec 4	CNN	10.3
	34	Wed	Dec 6	Unsupervised Learning & Clustering	12.1, 12.4
	35	Fri	Dec 8	Virtual: Project office hours	Project due