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

Announcements

Last time:

- 8.1 Decision Trees

This lecture:

- 8.2.1 Bagging
- 8.2.2 Random forest

Announcements:

- Some rearranging:

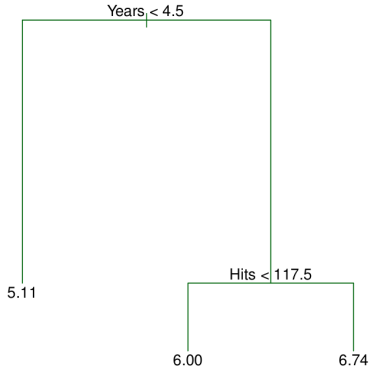
20	F	Nov 4	Polynomial & Step Functions.	7.1,7.2	
21	M	Nov 7	Step Functions	7.2	
22	W	Nov 9	Basis functions, Regression Splines	7.3,7.4	
23	F	Nov 11	Decision Trees	8.1	HW #7 Due
24	M	Nov 14	Random Forests	8.2.1, 8.2.2	
25	W	Nov 16	Maximal Margin Classifier	9.1	
26	F	Nov 18	SVC	9.2	HW #8 Due
27	M	Nov 21	SVM	9.3, 9.4, 9.5	
28	W	Nov 23	Extended virtual office hours		
	F	Nov 25	No class - Thanksgiving		
29	M	Nov 28	Single layer NN	10.1	HW #9 Due
30	W	Nov 30	Multi Layer NN	10.2	
31	F	Dec 2	CNN	10.3	
32	M	Dec 5	Unsupervised Learning & Clustering	12.1, 12.4	HW #10 Due
	W	Dec 7	Review		
	F	Dec 9	Midterm #3	Bring your cheat sheet and a non-internet-connected calculator	

Section 1

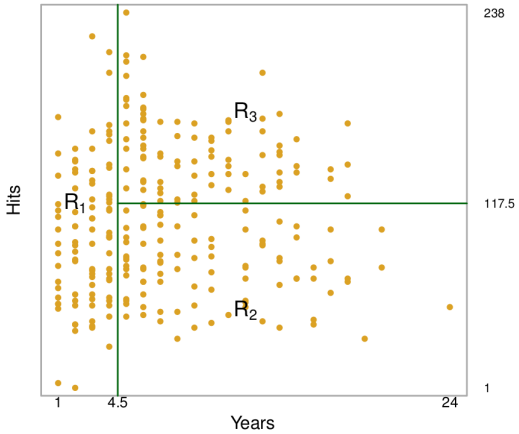
Last time

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

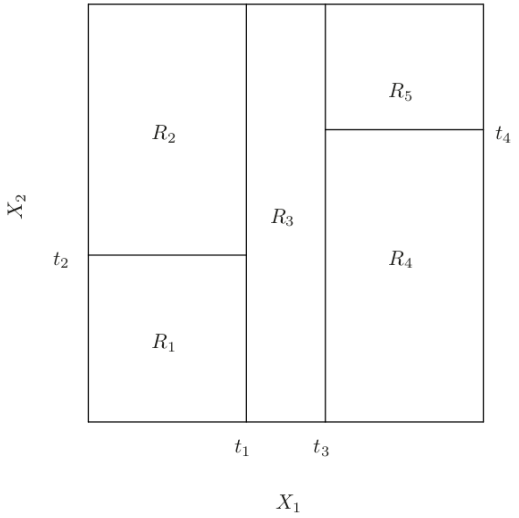


Viewing Regions Defined by Tree



How do we actually get the tree? Two steps

1. We divide the predictor space – that is, the set of possible values for $X_1, X_2, \cdots, X_p$ — into J distinct and non-overlapping regions, $R_1, R_2, \cdots, R_J$.
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 $R_1, \cdots, R_J$ that minimize

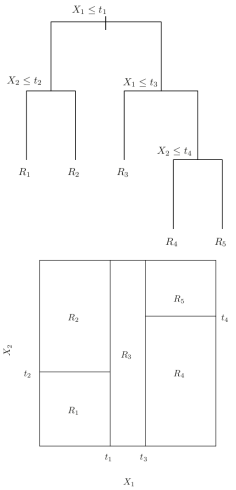
$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

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

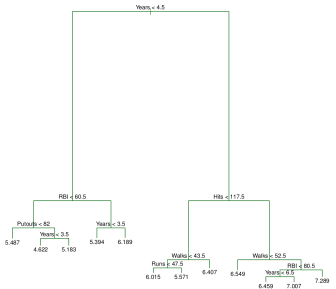
Pick s so that splitting into $\{X \mid X_j < s\}$ and $\{X \mid X_j \geq s\}$ results in largest possible reduction in RSS:

$$R_1(j, s) = \{X \mid X_j < s\}$$
$$R_2(j, s) = \{X \mid X_j \geq s\}$$

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

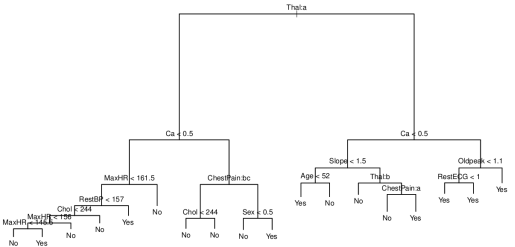


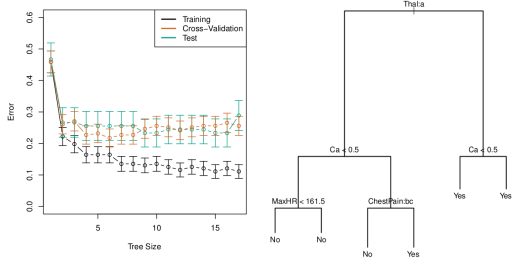
Pruning


Years < 4.5
RBI < 60.5
Hits < 117.5
Putouts < 82
Years < 3.5
Years < 3.5
Walks < 43.5
Walks < 52.5
Runs < 47.5
RBI < 60.5
Years < 6.5
5.487
4.622
5.183
5.394
6.189
6.015
5.571
6.407
6.549
6.459
7.007
7.289


Years < 4.5
5.11
Hits < 117.5
6.00
6.74

Classification version



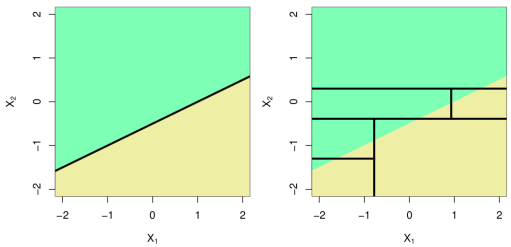


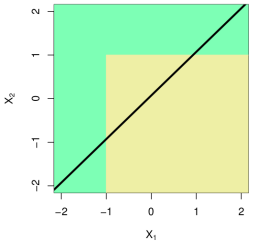
Evaluating the splits:

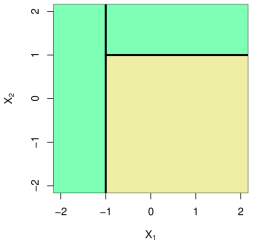
- $\hat{p}_{mk}$ = proportion of training observations in R_m from the kth class
- Error: $E = 1 - \max_k(\hat{p}_{mk})$
- Gini index:

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

Linear models vs trees







Pros/Cons

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:

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

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

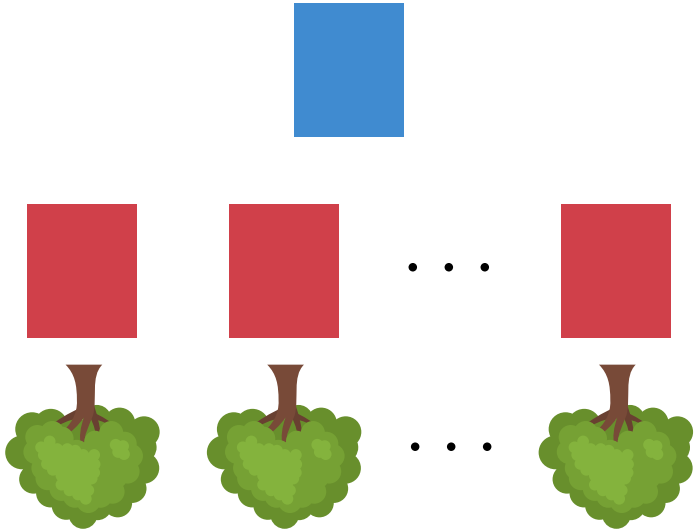
Boostrap modification:

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

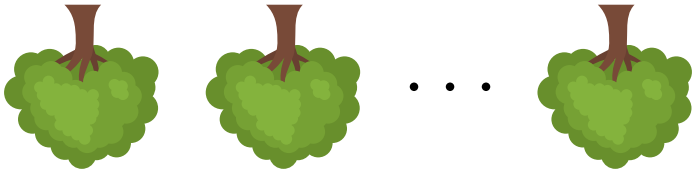
$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

or majority vote (classification)

Tree version



Prediction on new data point

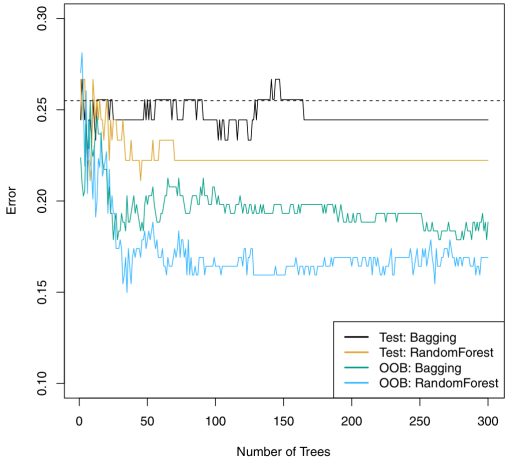


Bagging vs Bootstrap

Bootstrap

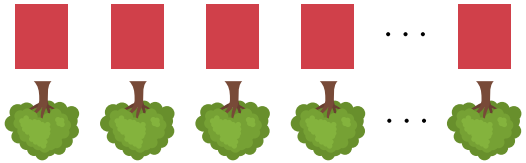
Bagging

Example: Heart classification data

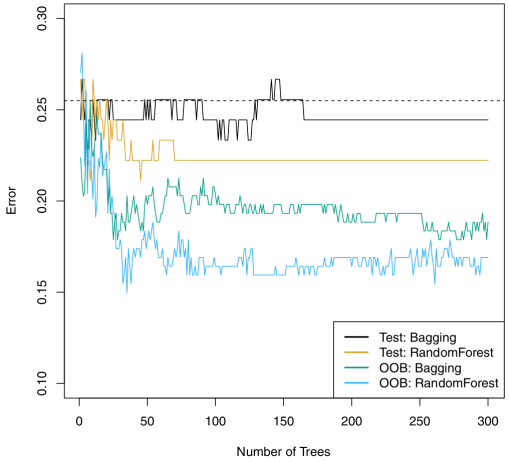


Out of Bag Error Estimation

- 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

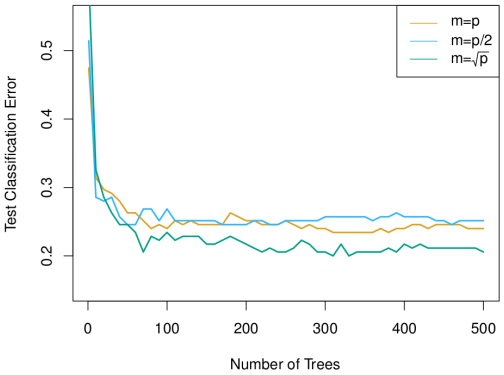
Random Forests

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

TL:DR

Next time

20	F	Nov 4	Polynomial & Step Functions.	7.1,7.2	
21	M	Nov 7	Step Functions	7.2	
22	W	Nov 9	Basis functions, Regression Splines	7.3,7.4	
23	F	Nov 11	Decision Trees	8.1	HW #7 Due
24	M	Nov 14	Random Forests	8.2.1, 8.2.2	
25	W	Nov 16	Maximal Margin Classifier	9.1	
26	F	Nov 18	SVC	9.2	HW #8 Due
27	M	Nov 21	SVM	9.3, 9.4, 9.5	
28	W	Nov 23	Extended virtual office hours		
	F	Nov 25	No class - Thanksgiving		
29	M	Nov 28	Single layer NN	10.1	HW #9 Due
30	W	Nov 30	Multi Layer NN	10.2	
31	F	Dec 2	CNN	10.3	
32	M	Dec 5	Unsupervised Learning & Clustering	12.1, 12.4	HW #10 Due
	W	Dec 7	Review		
	F	Dec 9	Midterm #3	Bring your cheat sheet and a non-internet-connected calculator	