

Ch 8.2.1, 8.2.2: Bagging and Random Forests

Lecture 25 - CMSE 381

Michigan State University

::

Dept of Computational Mathematics, Science & Engineering

Fri, Mar 27, 2026

Announcements

Last time:

- 8.1 Decision Trees - regression
- 8.1 Decision Trees - classification

This lecture:

- 8.2.1 Bagging
- 8.2.2 Random forest

Announcements:

- Homework 5 Due Sunday

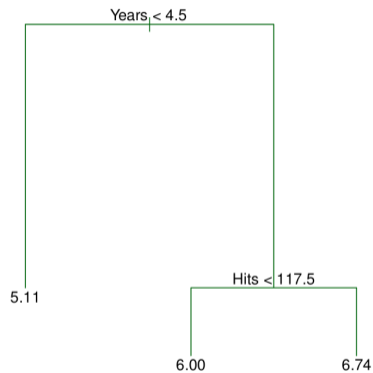
21	W	3/18	Polynomial & Step Functions	7.1-7.2		
22	F	3/20	Step Functions; Basis functions; Start Splines	7.2-7.4		
23	M	3/23	Regression Splines	7.4		
24	W	3/25	Decision Trees	8.1		Q7
25	F	3/27	Random Forests	8.2.1, 8.2.2	HW #5 Due Sun 3/29	
26	M	3/30	Maximal Margin Classifier	9.1		Q8
27	W	4/1	SVC	9.2		
28	F	4/3	SVM	9.3, 9.4		
29	M	4/6	Single Layer NN	10.1		
30	W	4/8	Multi Layer NN	10.2		Q9
31	F	4/10	CNN	10.3		
32	M	4/13	Unsupervised learning / clustering	12.1, 12.4	HW #6 Due Sun 4/12	
33	W	4/15	Virtual: Project Office Hours			Q10
	F	4/17	Review			
	M	4/20	Midterm #3			
	W	4/22				
	F	4/24				Project Due

Section 1

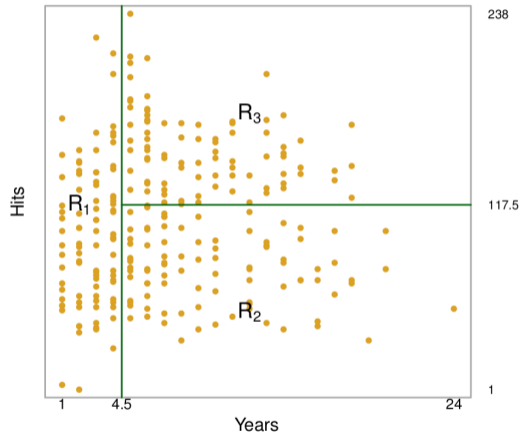
Previously

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

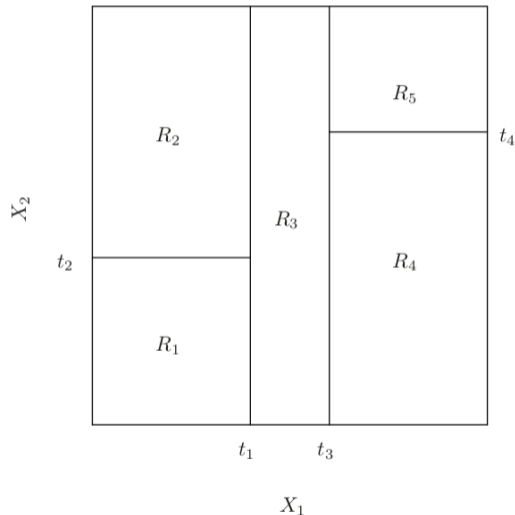


Viewing Regions Defined by Tree



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

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

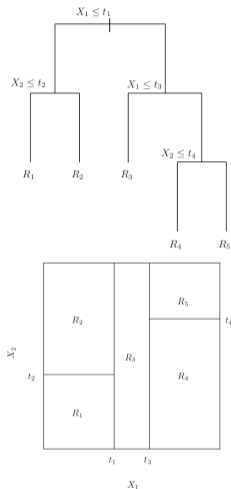
\hat{y}_{R_j} = mean response for training observations in j th box

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:

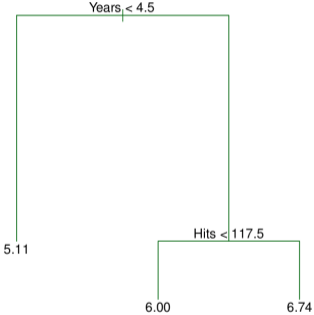
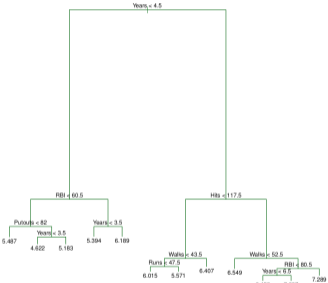
$$R_1(j, s) = \{X \mid X_j < s\}$$

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

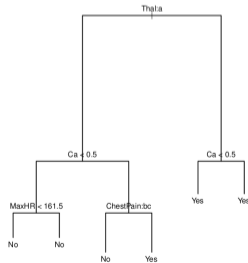
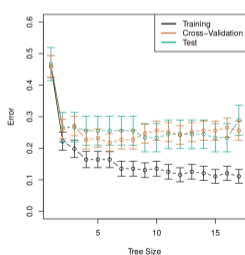
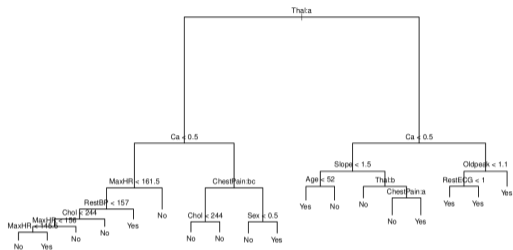
$$\sum_{i \mid x_i \in R_1(j, s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i \mid x_i \in R_2(j, s)} (y_i - \hat{y}_{R_2})^2$$



Pruning



Classification version

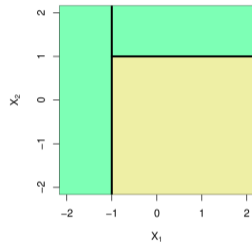
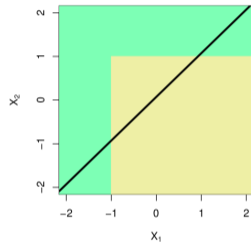
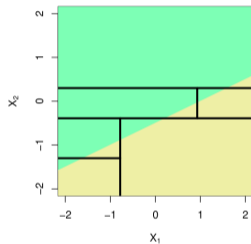
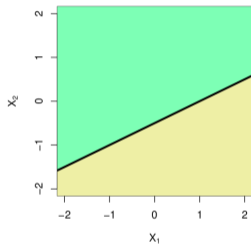


Evaluating the splits:

- \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



What will you learn today?

- What does bagging of decision trees accomplish?
- How do you use out of bag error estimation for decision trees?
 - ▶ You should be able to describe this for both regression and classification trees.
- What problem of bagging does using random forest address?
- What is the relationship between random forest and bagging?

Section 2

8.2.1 Bagging

Use ensemble of trees to reduce variance

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

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

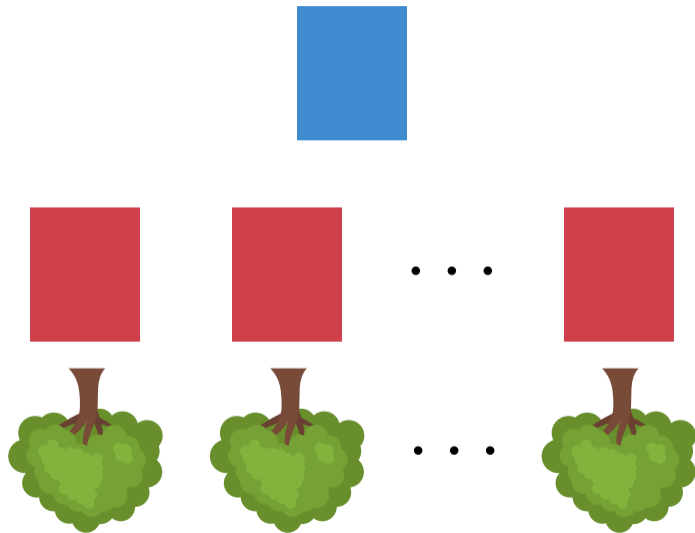
Bootstrap modification:

- 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)

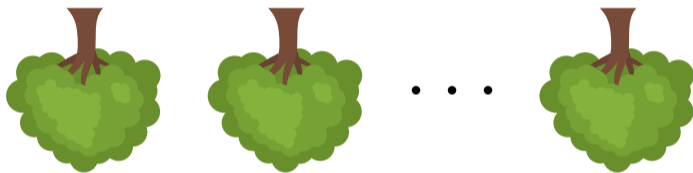
$$\hat{f}_{bag}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^{*b}(x)$$

or majority vote (classification)

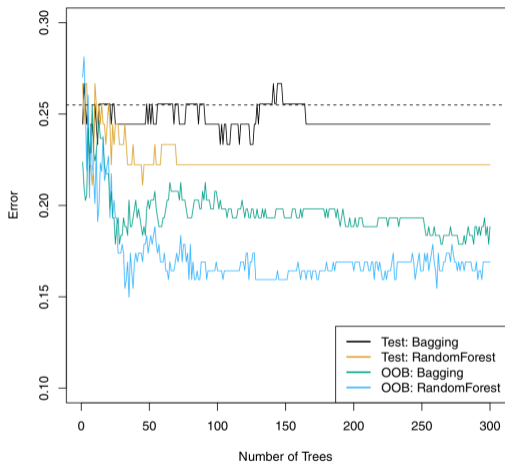
Tree version



Prediction on new data point

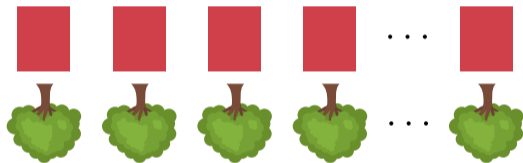


Example: Heart classification data

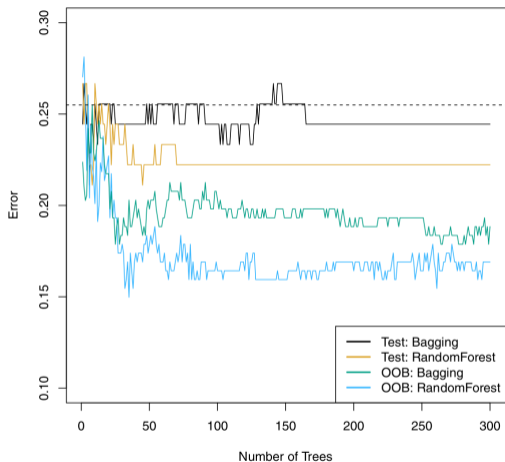


Out of Bag Error Estimation

- 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! rather law of large number
- Bootstrapped version of LOOCV.



Error using OOB



Test your understanding: [PolIEv](#)

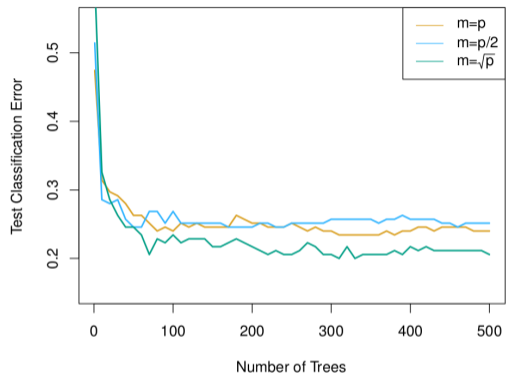
Section 3

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 forest 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 time!

- Bagging: trees grown independently on random samples. Trees tend to be similar to each other, can result in getting caught in local optima
- Random forest: trees independently on samples, but split is done using random subset of features

Next time

21	W	3/18	Polynomial & Step Functions	7.1-7.2		
22	F	3/20	Step Functions; Basis functions; Start Splines	7.2-7.4		
23	M	3/23	Regression Splines	7.4		Q7
24	W	3/25	Decision Trees	8.1		
25	F	3/27	Random Forests	8.2.1, 8.2.2	HW #5 Due Sun 3/29	
26	M	3/30	Maximal Margin Classifier	9.1		Q8
27	W	4/1	SVC	9.2		
28	F	4/3	SVM	9.3, 9.4		
29	M	4/6	Single Layer NN	10.1		Q9
30	W	4/8	Multi Layer NN	10.2		
31	F	4/10	CNN	10.3	HW #6 Due Sun 4/12	
32	M	4/13	Unsupervised learning / clustering	12.1, 12.4		Q10
33	W	4/15	Virtual: Project Office Hours			
	F	4/17	Review			
	M	4/20	Midterm #3			
	W	4/22				
	F	4/24			Project Due	