Supervised Machine Learning and Learning Theory

Lecture 10: Random forests, boosting

October 8, 2024

Warm-up questions

- Could you explain why LASSO can perform hard thresholding on small model coefficients?
- How does the ridge penalty affect model coefficients compared to OLS?
- When we apply LASSO/ridge, do we apply them to the intercept term or not? Explain the reason for that
- What's the key idea behind building a decision tree?

Warm-up questions

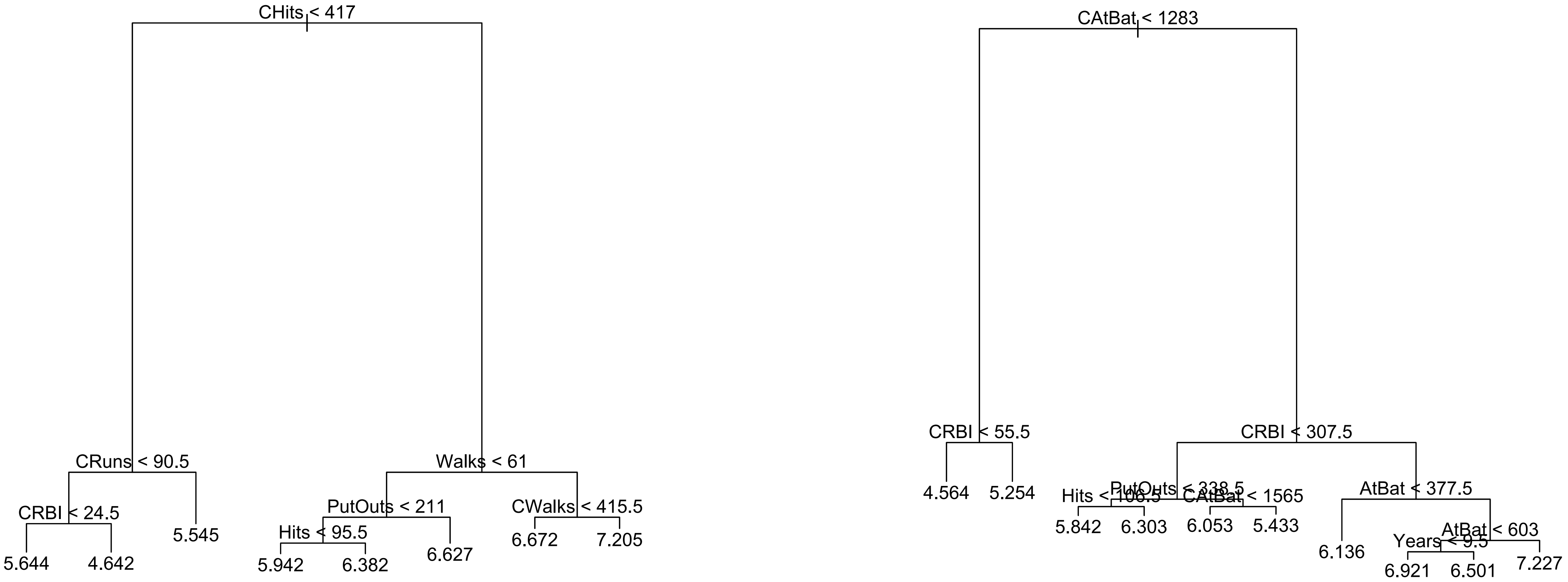
- How could we reduce the number of terminal nodes in a decision tree?
- Explain the high-level idea behind bootstrap

Lecture plan

- **Bagging**

Motivation

- **Example**: Predicting a baseball player's salary; split the training data into two equal-sized parts at random creates disparity



Bagging

- **Bootstrap aggregation**: Bagging is a way to reduce such variance
- **Example**: Estimate the mean of Z

Z_1	1.03
Z_2	1.56
Z_3	2.37
Z_4	2.13
Z_5	2.47

$$\bar{Z} = 1.91$$

$$\mathrm{Var}(\bar{Z}) = \frac{\sigma^2}{n} = \frac{1}{5} = 0.2$$

Data generating process: $Z \sim N(2,1)$

Toy example

- Suppose we have many independent sampling of datasets

Dataset 1

$Z_1^{(1)}$	1.03
$Z_2^{(1)}$	1.56
$Z_3^{(1)}$	2.37
$Z_4^{(1)}$	2.13
$Z_5^{(1)}$	2.47

$\bar{Z}^{(1)} = 1.91$
$\mathrm{Var}(\bar{Z}^{(1)}) = 0.2$

Dataset 2

$Z_1^{(2)}$	3.44
$Z_2^{(2)}$	3.06
$Z_3^{(2)}$	2.42
$Z_4^{(2)}$	2.40
$Z_5^{(2)}$	-0.78

$\bar{Z}^{(2)} = 2.11$
$\mathrm{Var}(\bar{Z}^{(2)}) = 0.2$

Dataset 3

$Z_1^{(3)}$	-0.13
$Z_2^{(3)}$	2.28
$Z_3^{(3)}$	2.09
$Z_4^{(3)}$	2.72
$Z_5^{(3)}$	1.40

$\bar{Z}^{(3)} = 1.67$
$\mathrm{Var}(\bar{Z}^{(3)}) = 0.2$

Dataset 4

$Z_1^{(4)}$	0.94
$Z_2^{(4)}$	1.84
$Z_3^{(4)}$	1.92
$Z_4^{(4)}$	2.49
$Z_5^{(4)}$	2.37

$\bar{Z}^{(4)} = 1.91$
$\mathrm{Var}(\bar{Z}^{(4)}) = 0.2$

$$\bar{Z}_{agg} = (\bar{Z}^{(1)} + \bar{Z}^{(2)} + \bar{Z}^{(3)} + \bar{Z}^{(4)})/4 = 1.90$$

$$\mathrm{Var}\left(\bar{Z}_{agg}\right) = \frac{0.2}{4} = 0.05$$

Toy example

- In practice, we only have one training dataset
- How can we create many datasets?

Z_1	1.03
Z_2	1.56
Z_3	2.37
Z_4	2.13
Z_5	2.47

Sampling with replacement →

Sample #1

Z_1	1.03
Z_2	1.56
Z_1	1.03
Z_5	2.47
Z_4	2.13

Sample #2

Z_4	2.13
Z_1	1.03
Z_3	2.37
Z_2	1.56
Z_3	2.37

Sample #3

Z_5	2.47
Z_2	1.56
Z_3	2.37
Z_2	1.56
Z_1	1.03

Sample #4

Z_5	2.47
Z_3	2.37
Z_3	2.37
Z_1	1.03
Z_2	1.56

Toy example

- Estimate the mean on each bootstrap sampling set

Sample #1

Z_1	1.03
Z_2	1.56
Z_5	2.47
Z_5	2.47
Z_4	2.13

$\bar{Z}^{(1)} = 1.93$

Sample #3

Z_5	2.47
Z_2	1.56
Z_3	2.37
Z_2	1.56
Z_1	1.03

$\bar{Z}^{(3)} = 1.80$

Sample #2

Z_4	2.13
Z_1	1.03
Z_3	2.37
Z_2	1.56
Z_3	2.37

$\bar{Z}^{(2)} = 1.89$

Sample #4

Z_5	2.47
Z_3	2.37
Z_3	2.37
Z_1	1.03
Z_2	1.56

$\bar{Z}^{(4)} = 1.96$

Toy example

- Average all estimates

$\bar{Z}^{(1)} = 1.93$ $\bar{Z}^{(2)} = 1.89$ $\bar{Z}^{(3)} = 1.80$ $\bar{Z}^{(4)} = 1.96$

$$\bar{Z}_{bag} = (\bar{Z}^{(1)} + \bar{Z}^{(2)} + \bar{Z}^{(3)} + \bar{Z}^{(4)})/4 = 1.90$$

- This is called **bagging** (**b**ootstrap **agg**regat**ing**)

- Bagging amounts to averaging the fits from B independent datasets, which would reduce the variance by a factor $\frac{1}{B}$

Bagging for decision trees

- Estimate a decision tree model $f(x)$ using bootstrap

X_1	Y_1
X_2	Y_2
X_3	Y_3
X_4	Y_4
X_5	Y_5

Sampling with replacement →

Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

Sample #4

X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

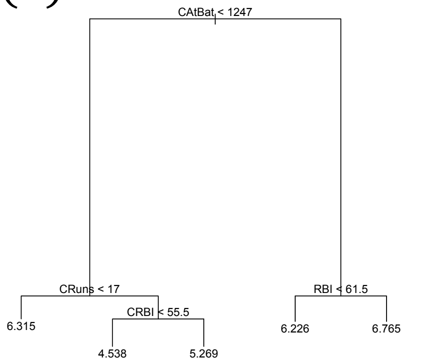
Bagging for decision trees

- Estimate a decision tree model $f(x)$ using bootstrap

Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

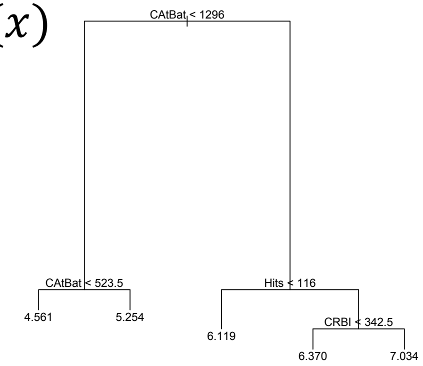
$\hat{f}^1(x)$



Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

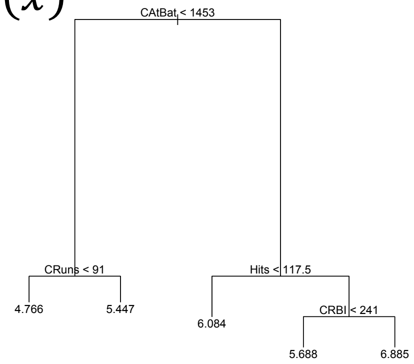
$\hat{f}^2(x)$



Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

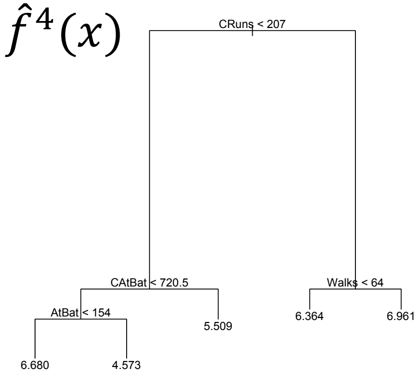
$\hat{f}^3(x)$



Sample #4

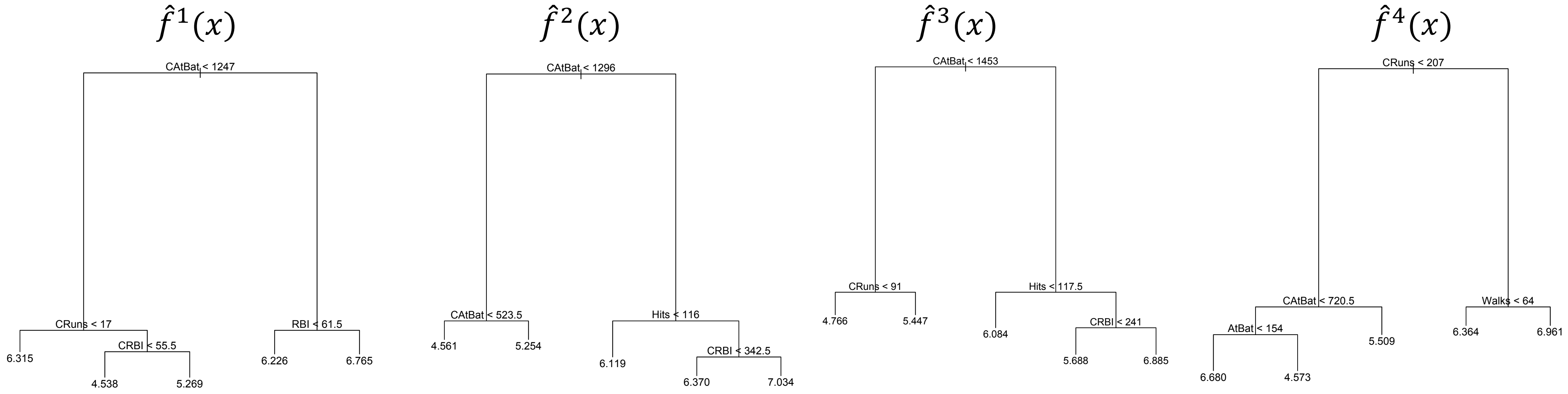
X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

$\hat{f}^4(x)$



Bagging for decision trees

- Average all the predictions



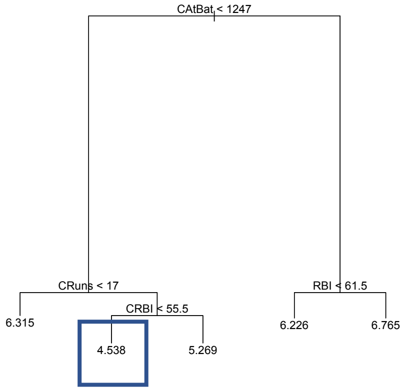
$$\hat{f}_{bag}(x) = \frac{1}{4}\left(\hat{f}^1(x) + \hat{f}^2(x) + \hat{f}^3(x) + \hat{f}^4(x)\right)$$

- If we have B bootstrapped samples, $\hat{f}_{bag}(x) = \frac{1}{B}\left(\hat{f}^1(x) + \hat{f}^2(x) + \cdots + \hat{f}^B(x)\right)$

Example

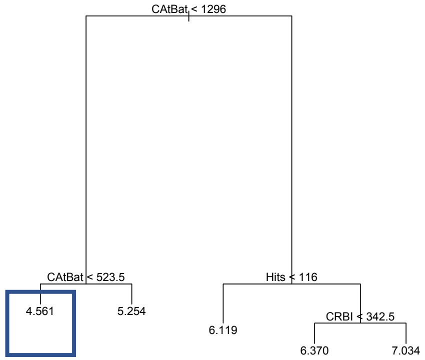
	AtBat	Hits	HmRun	Runs	RBI	Walks	Years	CAtBat	CHits	CHmRun	CRuns	CRBI	CWalks	League	Division	PutOuts	Assists	Errors	Salary	NewLeague
-Andy Allanson	293	66	1	30	29	14	1	293	66	1	30	29	14	A	E	446	33	20	NA	A

$\hat{f}^1(x)$



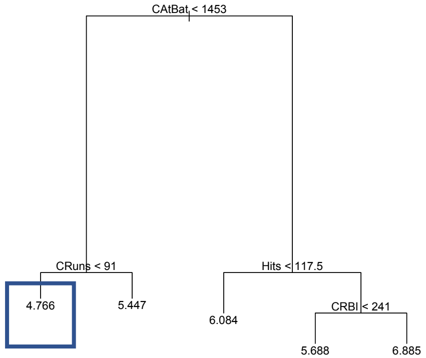
$\hat{f}^1(x) = 4.538$

$\hat{f}^2(x)$



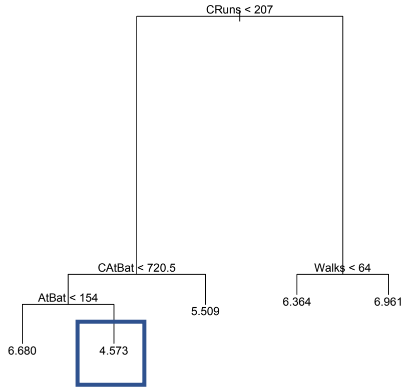
$\hat{f}^2(x) = 4.561$

$\hat{f}^3(x)$



$\hat{f}^3(x) = 4.766$

$\hat{f}^4(x)$



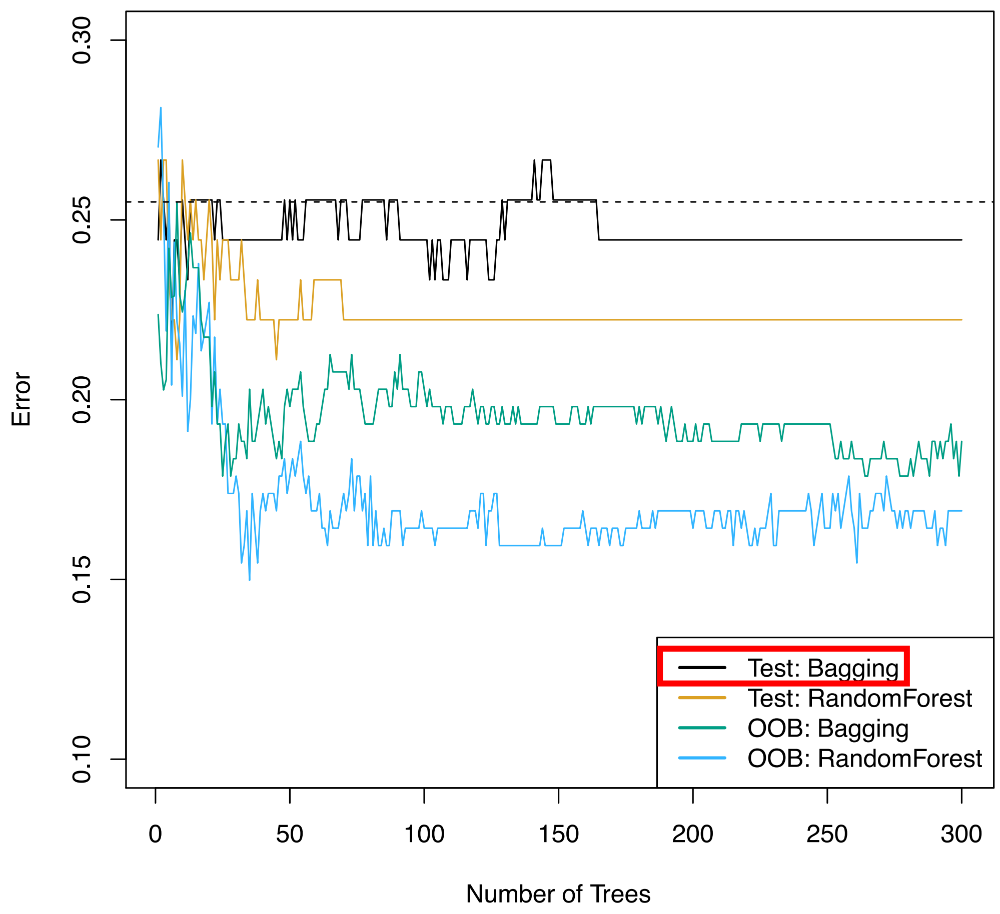
$\hat{f}^4(x) = 4.573$

$$\hat{f}_{bag}(x) = \frac{1}{4}(\hat{f}^1(x) + \hat{f}^2(x) + \hat{f}^3(x) + \hat{f}^4(x)) = (4.538 + 4.561 + 4.766 + 4.573)/4 = 4.6095$$

If the problem is classification, how would we aggregate the predictions?

Example

- **Example:** Predict whether a patient with chest pain has heart disease based on age, cholesterol, etc
- Dash line: Single tree
- Bagging outperforms a single decision tree
- The number of trees B does not matter after some threshold (in practice, $B = 100$ is sufficient when error has converged)



Cross-validation

- **Cross-validation:** To estimate the test error of a bagging estimate. How should we perform cross-validation with bootstrap?
  - Each time we draw a bootstrap sample, we only use 63% of the observations
  - Use the rest of the observations as a **holdout set**

Out-of-bag error

- **Idea**: Use the rest of the observations as a **holdout set**
  - **Out-of-bag (OOB) error:** For each sample X_i, find the prediction $\hat{Y}_i^b$ for all bootstrap samples b which do not contain X_i
  - Around $0.37B$ of them. Average these predictions to obtain $\hat{Y}_i^{oob}$
- **Example**: For the observation X_4, predict $\hat{Y}_4^b$

$$\hat{Y}_4^{oob} = \frac{1}{2}(\hat{Y}_4^3 + \hat{Y}_4^4)$$

$\hat{Y}_4^1$
Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

$\hat{Y}_4^2$
Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

$\hat{Y}_4^3$
Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

$\hat{Y}_4^4$
Sample #4

X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

Out-of-bag error

- **Step 1:** For each sample X_i, find the prediction $\hat{Y}_i^b$ for all bootstrap samples b which do not contain X_i. These should be around $0.37B$ of them. Average to obtain $\hat{Y}_i^{oob}$
- **Step 2:** Compute the error $(Y_i - \hat{Y}_i^{oob})^2$
- **Step 3:** Average the errors over all observations $i = 1, \cdots, n$
- **Example:** $\frac{1}{5}((Y_1 - \hat{Y}_1^{oob})^2 + (Y_2 - \hat{Y}_2^{oob})^2 + \cdots + (Y_5 - \hat{Y}_5^{oob})^2)$

Sample #1

X_1	Y_1
X_2	Y_2
X_1	Y_1
X_5	Y_5
X_4	Y_4

Sample #2

X_4	Y_4
X_1	Y_1
X_3	Y_3
X_2	Y_2
X_3	Y_3

Sample #3

X_5	Y_5
X_2	Y_2
X_3	Y_3
X_2	Y_2
X_1	Y_1

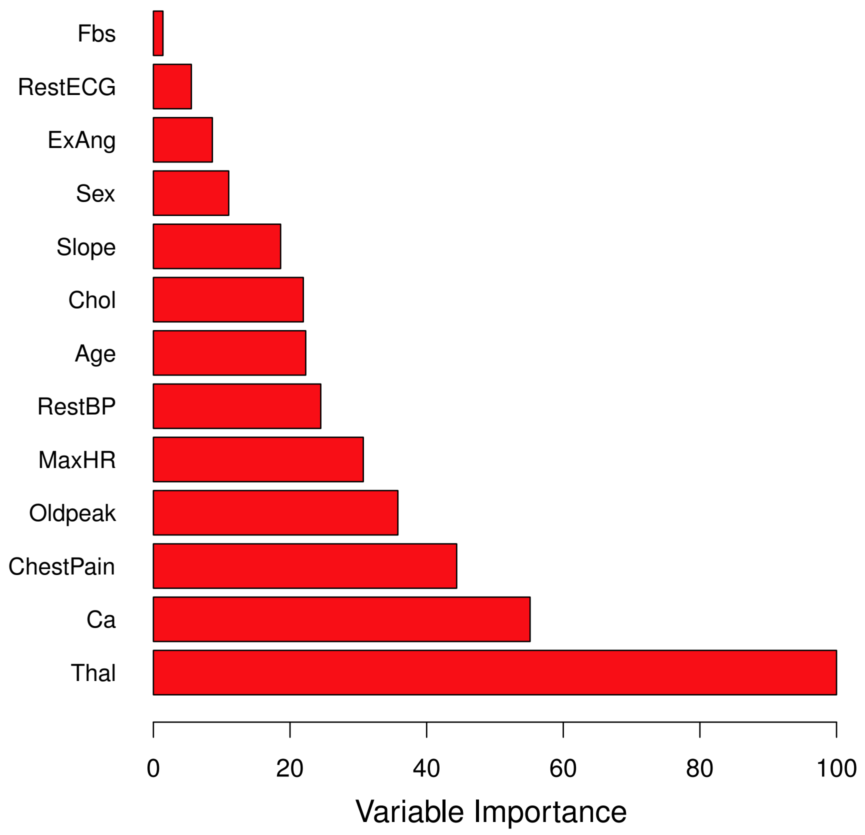
Sample #4

X_5	Y_5
X_3	Y_3
X_3	Y_3
X_1	Y_1
X_2	Y_2

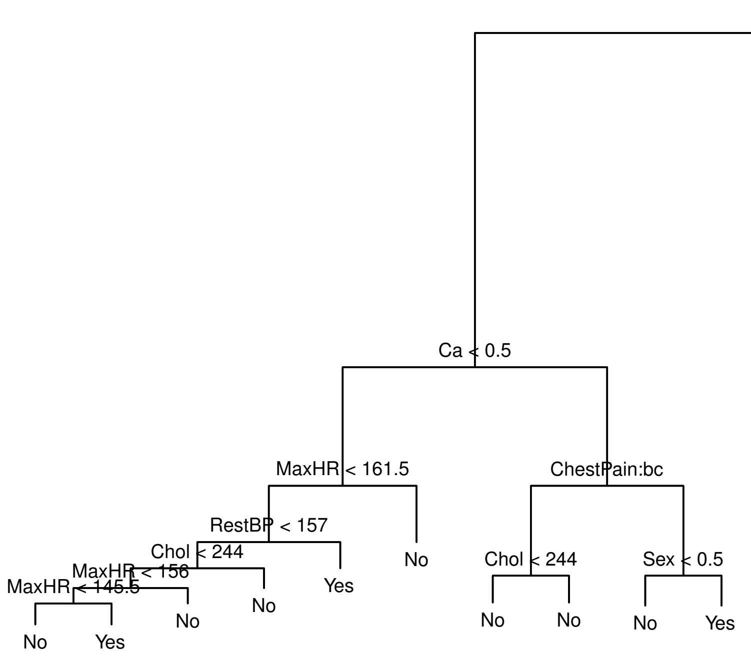
Feature importance

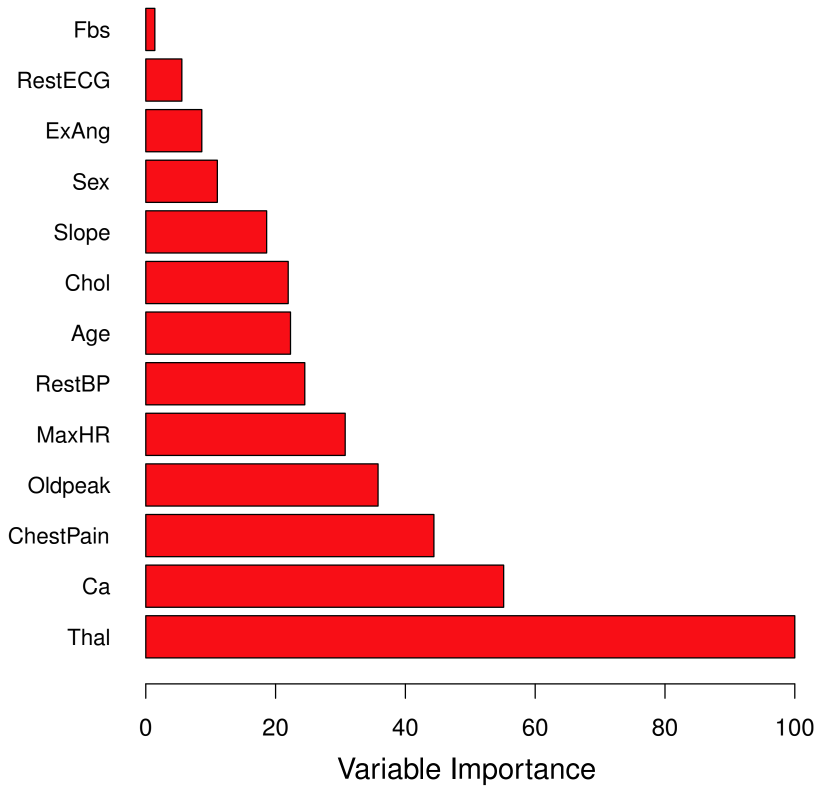
- For each predictor, add up the total amount by which the RSS (or Gini index) decreases in every split of the predictor
- Average the amount over all bootstrap estimates $T^1, \cdots, T^B$

- **Example:** Predicting heart disease



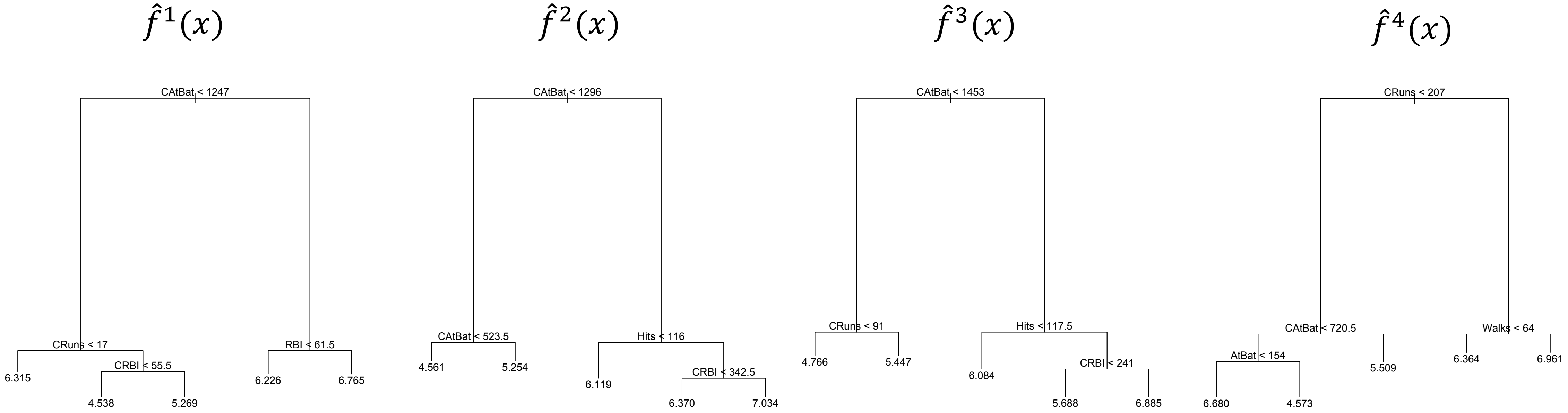
Feature importance





Bagging has a problem

- The trees produced by different bootstrap samples can be very similar: Three decision trees first split by CAtBat



Lecture plan

- **Random forest**

Random forests

- **Random forests: Bagging + random sampling of features**
  - Fit a decision tree with each bootstrap sample
  - To fit a tree, select a random subset of $m < p$ predictors to consider in each step
  - Lead to different trees from each sample
  - Finally, average the predictions of all trees

Random forests

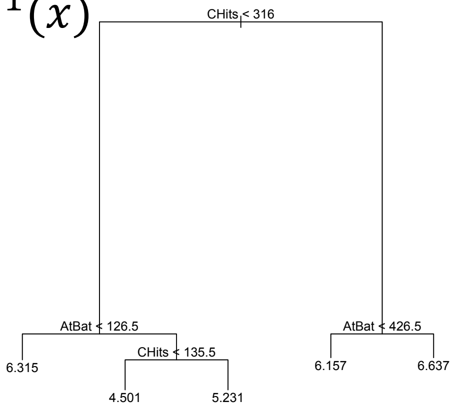
Random forests to predict a baseball player's salary: $p = 19$, $m = 5$

- $X_{i,j}$: jth predictor of observation i

Sample #1

$X_{1,4}$	$X_{1,17}$	$X_{1,9}$	$X_{1,6}$	$X_{1,1}$	Y_1
$X_{2,4}$	$X_{2,17}$	$X_{2,9}$	$X_{2,6}$	$X_{2,1}$	Y_2
$X_{1,4}$	$X_{1,17}$	$X_{1,9}$	$X_{1,6}$	$X_{1,1}$	Y_1
$X_{5,4}$	$X_{5,17}$	$X_{5,9}$	$X_{5,6}$	$X_{5,1}$	Y_5
$X_{4,4}$	$X_{4,17}$	$X_{4,9}$	$X_{4,6}$	$X_{4,1}$	Y_4

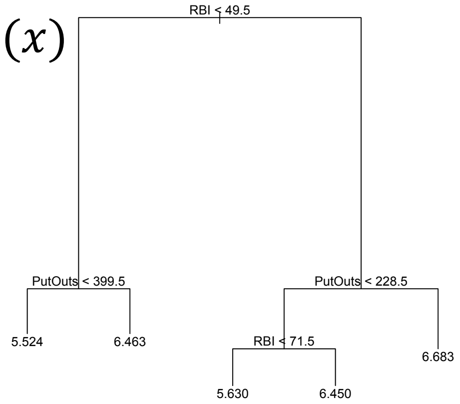
$\hat{f}^1(x)$



Sample #2

$X_{4,16}$	$X_{4,5}$	$X_{4,19}$	$X_{4,18}$	$X_{4,1}$	Y_4
$X_{1,16}$	$X_{1,5}$	$X_{1,19}$	$X_{1,18}$	$X_{1,1}$	Y_1
$X_{3,16}$	$X_{3,5}$	$X_{3,19}$	$X_{3,18}$	$X_{3,1}$	Y_3
$X_{2,16}$	$X_{2,5}$	$X_{2,19}$	$X_{2,18}$	$X_{2,1}$	Y_2
$X_{3,16}$	$X_{3,5}$	$X_{3,19}$	$X_{3,18}$	$X_{3,1}$	Y_3

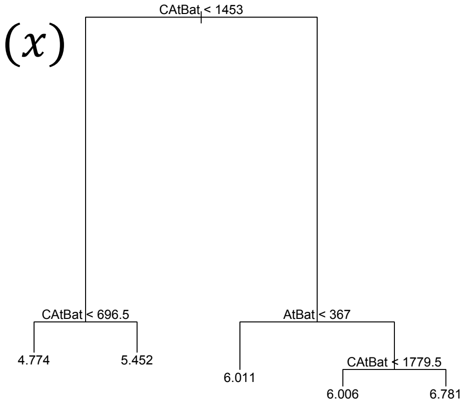
$\hat{f}^2(x)$



Sample #3

$X_{5,6}$	$X_{5,14}$	$X_{5,1}$	$X_{5,4}$	$X_{5,8}$	Y_5
$X_{2,6}$	$X_{2,14}$	$X_{2,1}$	$X_{2,4}$	$X_{2,8}$	Y_2
$X_{3,6}$	$X_{3,14}$	$X_{3,1}$	$X_{3,4}$	$X_{3,8}$	Y_3
$X_{2,6}$	$X_{2,14}$	$X_{2,1}$	$X_{2,4}$	$X_{2,8}$	Y_2
$X_{1,6}$	$X_{1,14}$	$X_{1,1}$	$X_{1,4}$	$X_{1,8}$	Y_1

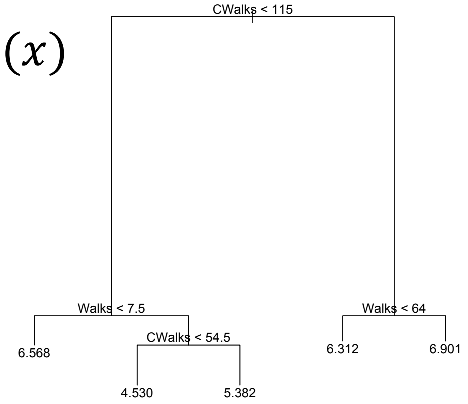
$\hat{f}^3(x)$



Sample #4

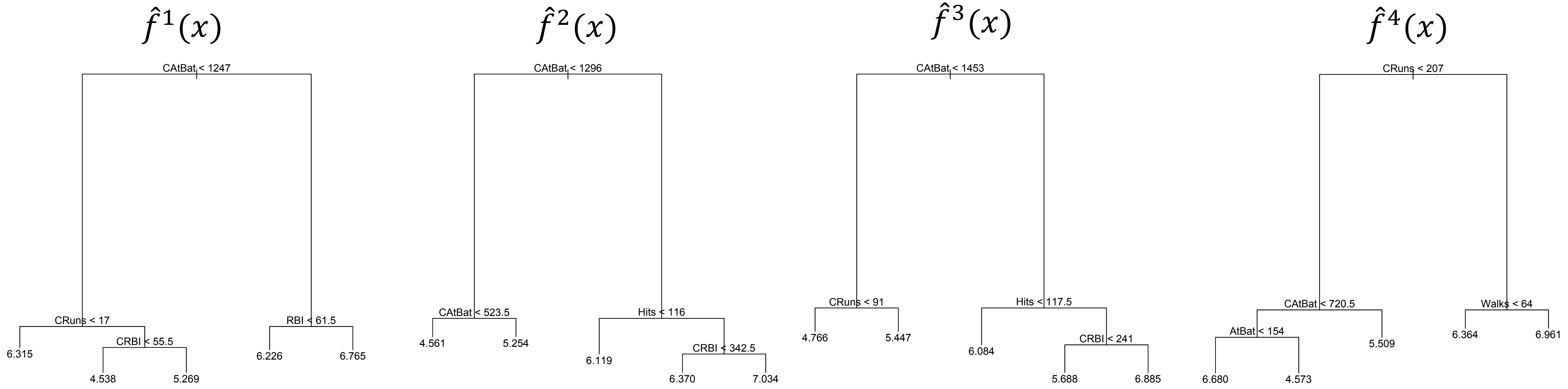
$X_{5,17}$	$X_{5,6}$	$X_{5,13}$	$X_{5,5}$	$X_{5,7}$	Y_5
$X_{3,17}$	$X_{3,6}$	$X_{3,13}$	$X_{3,5}$	$X_{3,7}$	Y_3
$X_{3,17}$	$X_{3,6}$	$X_{3,13}$	$X_{3,5}$	$X_{3,7}$	Y_3
$X_{1,17}$	$X_{1,6}$	$X_{1,13}$	$X_{1,5}$	$X_{1,7}$	Y_1
$X_{2,17}$	$X_{2,6}$	$X_{2,13}$	$X_{2,5}$	$X_{2,7}$	Y_2

$\hat{f}^4(x)$



Random forests

Average the predictions of all trees



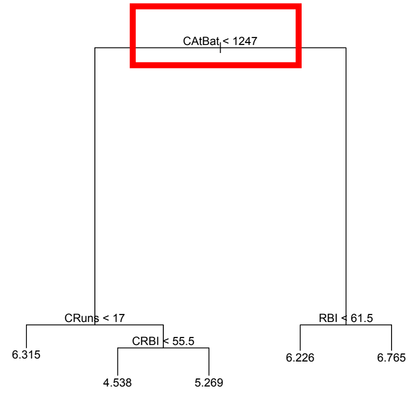
$$\hat{f}_{rf}(x) = \frac{1}{4}(\hat{f}^1(x) + \hat{f}^2(x) + \hat{f}^3(x) + \hat{f}^4(x))$$

More generally, if we have B bootstrapped training datasets, $\hat{f}_{rf}(x) = \frac{1}{B}(\hat{f}^1(x) + \hat{f}^2(x) + \cdots + \hat{f}^B(x))$

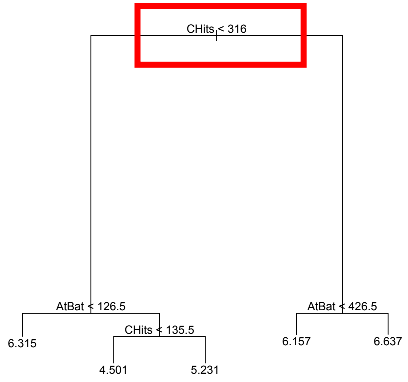
Bagging vs. random forests

$\hat{f}^1(x)$

Bagging

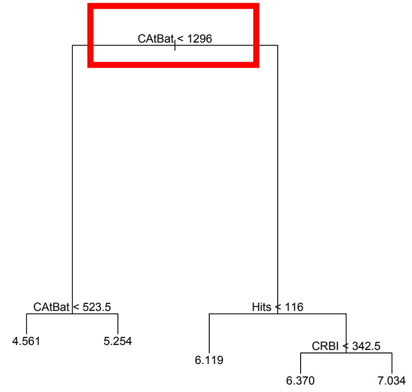


Random forests

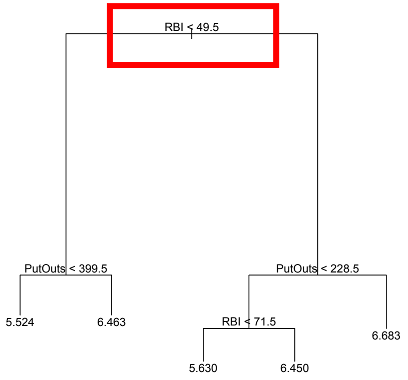


$\hat{f}^2(x)$

Bagging

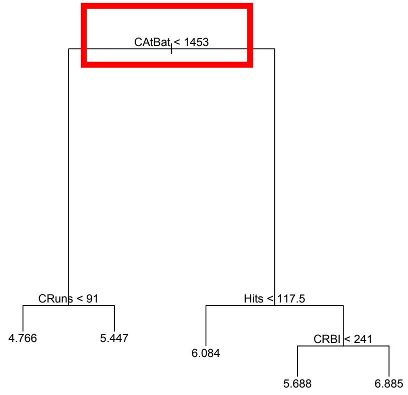


Random forests

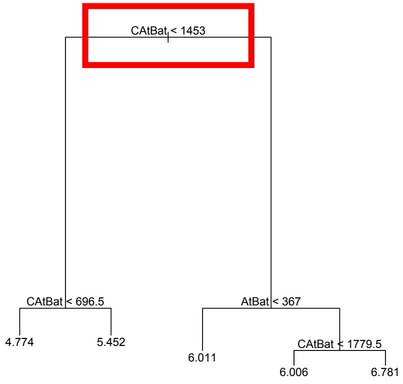


$\hat{f}^3(x)$

Bagging

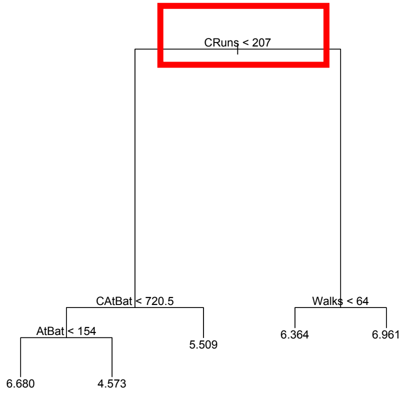


Random forests

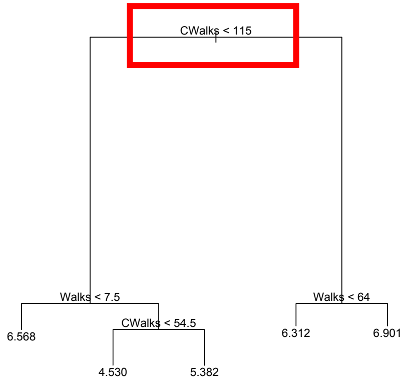


$\hat{f}^4(x)$

Bagging

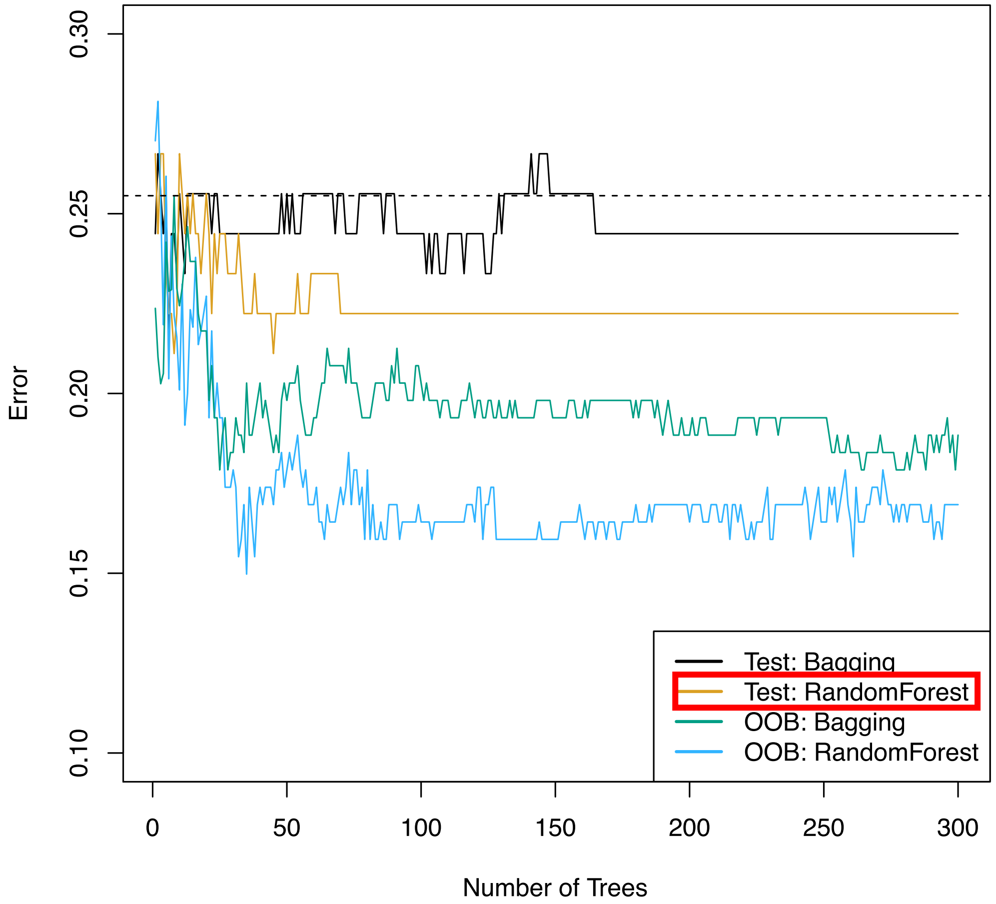


Random forests



Bagging vs. random forests

- Random forests outperform Bagging



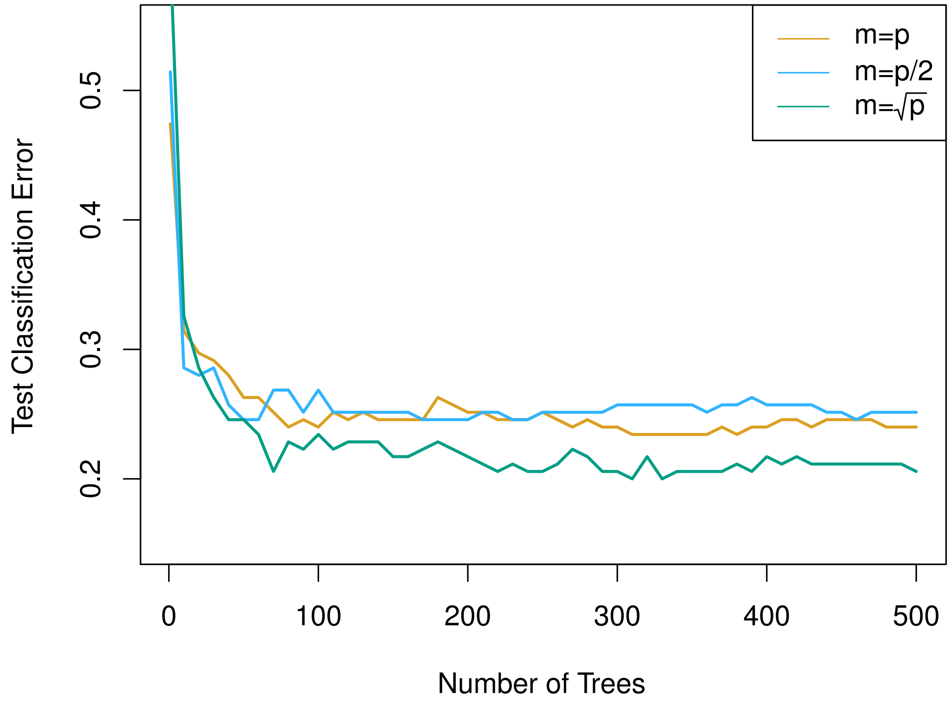
Choosing m in random forests

- **Example:** Predict cancer type (either normal or 1 of 14 different types of cancer) based on 500 genes

  - Error rate of a single tree: 45.6%

  - Using 400 trees is sufficient

  - m (# of features per tree) is a tuning parameter



Lecture plan

- **Gradient boosting**
- **AdaBoost**

Gradient boosting

- Random forests involve a lot of randomness and requires fitting many decision trees

- Gradient boosting uses less randomness
  - Trees are grown sequentially using the remaining features from previous trees
  - Each tree is fit on a modified version of the original data
  - Related to partial least squares

- Gradient boosting is also more scalable

Digress: Gradient descent

- An iterative approach to minimize a loss function

- Given a function f with parameters θ_t, at iteration t

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla f(\theta_t)$$

  - η is a learning rate
  - B is total number of iterations

Back to gradient boosting

- **Step 1**: Set $\hat{f}(x) = 0$, and $r_i = y_i$ for $i = 1, \cdots, n$

- **Step 2**: For $b = 1, \cdots, B$, iterate:
  - Fit a decision tree $\hat{f}^b$ with d splits to the response $r_1, \cdots, r_n$
  - Update the prediction to
$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$
  - Update the residuals
$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

- **Step 3**: Output the final model
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Gradient boosting

X_1	Y_1
X_2	Y_2
X_3	Y_3
X_4	Y_4
X_5	Y_5

$r_i^1 \leftarrow Y_i - \lambda \hat{f}^1(X_i)$

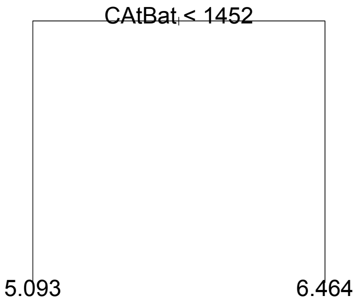
X_1	r_1^1
X_2	r_2^1
X_3	r_3^1
X_4	r_4^1
X_5	r_5^1

$r_i^2 \leftarrow r_i^1 - \lambda \hat{f}^2(X_i)$

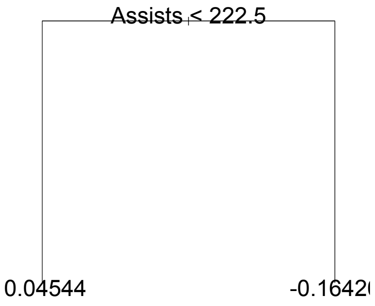
X_1	r_1^2
X_2	r_2^2
X_3	r_3^2
X_4	r_4^2
X_5	r_5^2

…

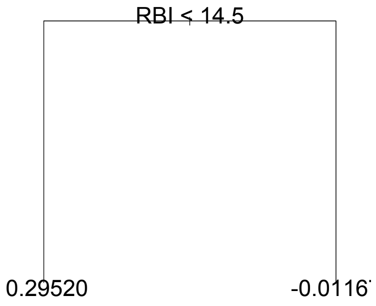
$\hat{f}^1(x)$



$\hat{f}^2(x)$



$\hat{f}^3(x)$



$$\hat{f}(x) = \lambda \hat{f}^1(x) + \lambda \hat{f}^2(x) + +\lambda \hat{f}^3(x) + \cdots + \lambda \hat{f}^B(x)$$

Hyperparameters

- The number of trees B
  - Boosting can overfit if B is too large (a.k.a. early stopping)
  - Use cross-validation to select B

- The learning rate λ
  - Typical values are 0.01 or 0.001
  - Very small λ requires a large B to achieve good performance

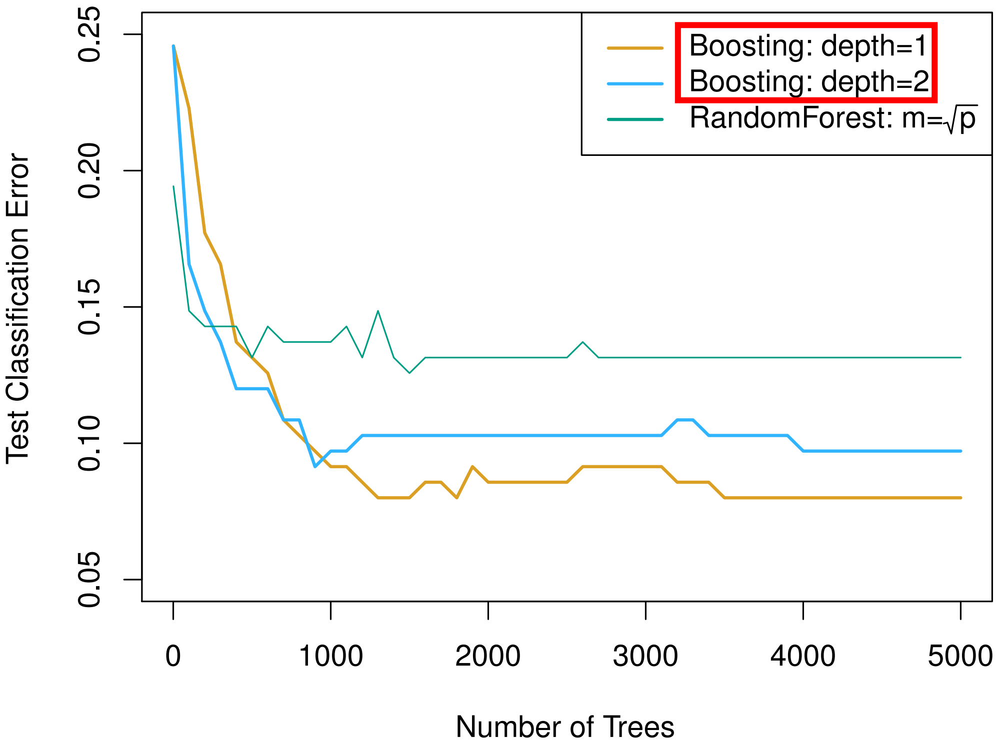
- The number of splits/depth d in each tree, e.g., $d = 1$

Boosting vs. random forests

Example: Predict cancer type (either normal or 1 of 14 different types of cancer) based on 500 genes; $\lambda = 0.01$

- Depth-1 trees outperform depth-2 trees

- Both outperform random forests



AdaBoost

- Training a boosted classifier
- For example, $Y \in \{-1,1\}$

Initial weight

X_1	Y_1	$1/5$
X_2	Y_2	$1/5$
X_3	Y_3	$1/5$
X_4	Y_4	$1/5$
X_5	Y_5	$1/5$

Fitted tree $\hat{f}^1(x)$
Correctly predict all samples besides Y_3 and Y_5

$$Error = \frac{1}{n}\sum_i I(\hat{f}(X_i) \neq Y_i) = \frac{2}{5}$$

$$\frac{1}{2}\log\frac{1 - Total\ Error}{Total\ Error} = \frac{1}{2}\log\frac{1-2/5}{2/5} = 0.088$$

Increase sample weight for the sample that was incorrectly classified
Decrease sample weight for the sample that was correctly classified

AdaBoost

- $Y \in \{-1,1\}$

Initial weight

X_1	Y_1	$1/5$
X_2	Y_2	$1/5$
X_3	Y_3	$1/5$
X_4	Y_4	$1/5$
X_5	Y_5	$1/5$

Fitted tree $\hat{f}^1(x)$
Correctly predict all samples besides Y_3 and Y_5

$$\frac{1}{2}\log\frac{1-Total\ Error}{Total\ Error} = \frac{1}{2}\log\frac{1-2/5}{2/5} = 0.088$$

Increase the sample weight for the sample that was incorrectly classified

$$New\ sample\ weight = sample\ weight \times \exp(Amount\ of\ stay)$$

$$New\ sample\ weight = \frac{1}{5} \times \exp(Amount\ of\ stay) = 0.2184$$

Decrease the sample weight for the sample that was correctly classified

$$New\ sample\ weight = sample\ weight \times \exp(-Amount\ of\ stay)$$

$$New\ sample\ weight = \frac{1}{5} \times \exp(-Amount\ of\ stay) = 0.1831$$

AdaBoost

Initial weight

X_1	Y_1	$1/5$
X_2	Y_2	$1/5$
X_3	Y_3	$1/5$
X_4	Y_4	$1/5$
X_5	Y_5	$1/5$

Update weight

New weight

X_1	Y_1	0.1831
X_2	Y_2	0.1831
X_3	Y_3	0.2184
X_4	Y_4	0.1831
X_5	Y_5	0.2184

Fitted tree $\hat{f}^1(x)$
Correctly predict all samples besides Y_3 and Y_5

$$Sum\ of\ the\ weights\ =\ 0.9862 \neq 1$$

AdaBoost

Initial weight

X_1	Y_1	$1/5$
X_2	Y_2	$1/5$
X_3	Y_3	$1/5$
X_4	Y_4	$1/5$
X_5	Y_5	$1/5$

Update weight

New weight

X_1	Y_1	0.1831/0.9862
X_2	Y_2	0.1831/0.9862
X_3	Y_3	0.2184/0.9862
X_4	Y_4	0.1831/0.9862
X_5	Y_5	0.2184/0.9862

…

Fitted tree $\hat{f}^1(x)$

Correctly predict all samples besides Y_3 and Y_5

Fitted tree $\hat{f}^2(x)$

Predict the most likely class: $\hat{f}(x) = Sign\left(\sum_{b=1}^{B} \lambda_b \hat{f}^b(x)\right)$

Announcements

- Reading material: Chapter 8 (in particular, Chapter 8.2/8.3), ISLP
- HW1 grading will be released by this Friday