Bagging and Random Forests

Prof Wells

STA 295: Stat Learning

April 23rd, 2024

Outline

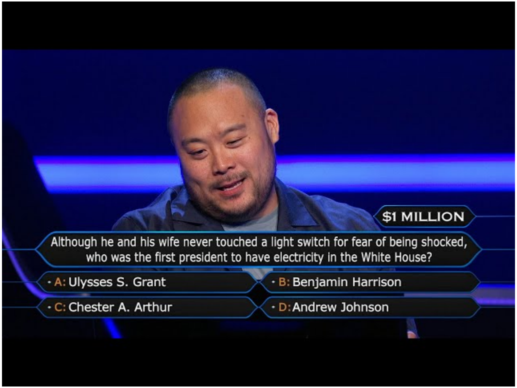
- Introduce ensemble modeling as means of improving low accuracy models
- Discuss bagging and random forests as methods for reducing variance in decision trees
- Implement random forests in R

Section 1

Ensemble Models

Who Wants to Be a Millionaire?

- *Who Wants to Be a Millionaire* is a television gameshow that debuted in the 1990s and in which contestants answer a series of increasingly difficult multiple choice questions in order to win the grand prize of $1,000,000.



$1 MILLION

Although he and his wife never touched a light switch for fear of being shocked, who was the first president to have electricity in the White House?

A: Ulysses S. Grant
B: Benjamin Harrison
C: Chester A. Arthur
D: Andrew Johnson

Who Wants to Be a Millionaire?

- The original show included 3 “lifeline” options contestants could use to answer questions:
  - **50:50**: Two randomly selected incorrect answers are eliminated
  - **Phone a Friend**: The contestant calls a friend and is given 30 seconds to discuss
  - **Ask the Audience**: Audience members each vote on the answer they think is correct

Who Wants to Be a Millionaire?

- The original show included 3 “lifeline” options contestants could use to answer questions:
  - **50:50**: Two randomly selected incorrect answers are eliminated
  - **Phone a Friend**: The contestant calls a friend and is given 30 seconds to discuss
  - **Ask the Audience**: Audience members each vote on the answer they think is correct

- Which lifeline has the highest chance of producing the correct answer?

Who Wants to Be a Millionaire?

- The original show included 3 "lifeline" options contestants could use to answer questions:
  - **50:50**: Two randomly selected incorrect answers are eliminated
  - **Phone a Friend**: The contestant calls a friend and is given 30 seconds to discuss
  - **Ask the Audience**: Audience members each vote on the answer they think is correct

- Which lifeline has the highest chance of producing the correct answer?

Why?

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?
- In fact, there are 28 grocery aisles.

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?
- In fact, there are 28 grocery aisles.
- A survey of 30 current STEM students in STA 209 were asked the same question.
    - The average value of their guess was 28.4, with standard deviation of 16.2

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?
- In fact, there are 28 grocery aisles.
- A survey of 30 current STEM students in STA 209 were asked the same question.
    - The average value of their guess was 28.4, with standard deviation of 16.2
- The rMSE for individual students was 15.9.

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?
- In fact, there are 28 grocery aisles.
- A survey of 30 current STEM students in STA 209 were asked the same question.
  - The average value of their guess was 28.4, with standard deviation of 16.2
- The rMSE for individual students was 15.9.
- However, the rMSE for the class was only 0.4.

Ensembling

At the Grinnell Walmart, consider the grocery section (south side of the store, to the left when you walk in). Write your answer to the following question on a sheet of paper. Do not consult your neighbor:

- How many grocery aisles are there?
- In fact, there are 28 grocery aisles.
- A survey of 30 current STEM students in STA 209 were asked the same question.
  - The average value of their guess was 28.4, with standard deviation of 16.2
- The rMSE for individual students was 15.9.
- However, the rMSE for the class was only 0.4.
- What factors about the class would cause such a drastic reduction in rMSE?

Ensemble Methods

- Suppose we have m different models to predict Y based on $X_1, \ldots, X_n$. Suppose $\hat{Y}_i$ is the prediction made by the ith model.

Ensemble Methods

- Suppose we have m different models to predict Y based on $X_1, \ldots, X_n$. Suppose $\hat{Y}_i$ is the prediction made by the ith model.
- A simple ensemble model makes a prediction $\hat{Y}$ as the weighted average of the predictions from each model:

$$\hat{Y} = w_1 \hat{Y}_1 + \cdots + w_m \hat{Y}_m \qquad \text{where } w_1 + \ldots w_m = 1, \quad w_i \geq 0$$

Ensemble Methods

- Suppose we have m different models to predict Y based on $X_1, \ldots, X_n$. Suppose $\hat{Y}_i$ is the prediction made by the ith model.
- A simple ensemble model makes a prediction $\hat{Y}$ as the weighted average of the predictions from each model:

$$\hat{Y} = w_1 \hat{Y}_1 + \cdots + w_m \hat{Y}_m \qquad \text{where } w_1 + \ldots w_m = 1, \quad w_i \geq 0$$

- Advantages of ensemble models?

Ensemble Methods

- Suppose we have m different models to predict Y based on $X_1, \ldots, X_n$. Suppose $\hat{Y}_i$ is the prediction made by the ith model.
- A simple ensemble model makes a prediction $\hat{Y}$ as the weighted average of the predictions from each model:

$$\hat{Y} = w_1 \hat{Y}_1 + \cdots + w_m \hat{Y}_m \qquad \text{where } w_1 + \ldots w_m = 1, \quad w_i \geq 0$$

- Advantages of ensemble models?
  - Significantly more flexible than a single model
  - More efficient than single model
  - More resilient against model-building bias

Ensemble Methods

- Suppose we have m different models to predict Y based on $X_1, \ldots, X_n$. Suppose $\hat{Y}_i$ is the prediction made by the ith model.
- A simple ensemble model makes a prediction $\hat{Y}$ as the weighted average of the predictions from each model:

$$\hat{Y} = w_1 \hat{Y}_1 + \cdots + w_m \hat{Y}_m \qquad \text{where } w_1 + \ldots w_m = 1, \quad w_i \geq 0$$

- Advantages of ensemble models?
    - Significantly more flexible than a single model
    - More efficient than single model
    - More resilient against model-building bias
- Disadvantages?

Ensemble Methods

- Suppose we have m different models to predict Y based on $X_1, \ldots, X_n$. Suppose $\hat{Y}_i$ is the prediction made by the ith model.
- A simple ensemble model makes a prediction $\hat{Y}$ as the weighted average of the predictions from each model:

$$\hat{Y} = w_1 \hat{Y}_1 + \cdots + w_m \hat{Y}_m \qquad \text{where } w_1 + \ldots w_m = 1, \quad w_i \geq 0$$

- Advantages of ensemble models?
  - Significantly more flexible than a single model
  - More efficient than single model
  - More resilient against model-building bias
- Disadvantages?
  - Making predictions is more computationally expensive
  - Favors models with low test time
  - Diminishing returns on the number models that can be incorporated in ensemble

Section 2

Bagging

Bootstrapping

- Many occasions where it would be useful to have multiple samples from a population:
  - Estimating standard errors of a statistic
  - Creating confidence intervals to estimate a parameter
  - Estimating variability in test RMSE

Bootstrapping

- Many occasions where it would be useful to have multiple samples from a population:
    - Estimating standard errors of a statistic
    - Creating confidence intervals to estimate a parameter
    - Estimating variability in test RMSE
- However, subsetting existing sample to create many smaller samples is problematic:
    - Diminishes ability to build accurate models, since less data is used
    - Variability in statistics depends on sample size

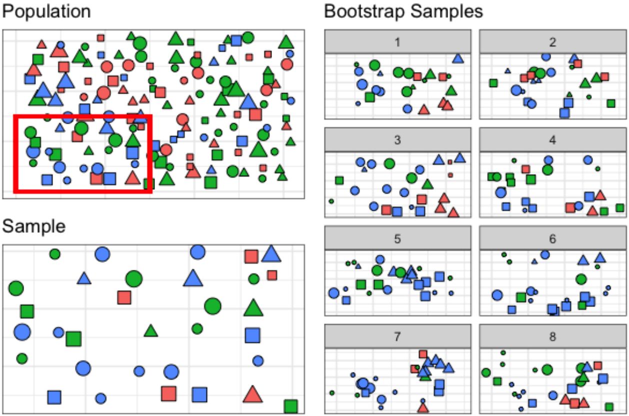
Bootstrapping

- Many occasions where it would be useful to have multiple samples from a population:
    - Estimating standard errors of a statistic
    - Creating confidence intervals to estimate a parameter
    - Estimating variability in test RMSE
- However, subsetting existing sample to create many smaller samples is problematic:
    - Diminishes ability to build accurate models, since less data is used
    - Variability in statistics depends on sample size
- Instead, we create **bootstrap** samples by sampling *with replacement* from the original sample a number of times equal to the original sample size

Bootstrapping

- Many occasions where it would be useful to have multiple samples from a population:
  - Estimating standard errors of a statistic
  - Creating confidence intervals to estimate a parameter
  - Estimating variability in test RMSE
- However, subsetting existing sample to create many smaller samples is problematic:
  - Diminishes ability to build accurate models, since less data is used
  - Variability in statistics depends on sample size
- Instead, we create **bootstrap** samples by sampling *with replacement* from the original sample a number of times equal to the original sample size
  - Since we sample with replacement, some observations from original sample are included more than once while others are omitted, which introduces a source of variability
  - Bootstrap sample is same size as original, and so can be used to estimate variability in statistic

Bootstrap Visualization



Bootstrap Sample Computations

Suppose we have a sample with 4 unique observations $\{x_1, x_2, x_3, x_4\}$ and create a bootstrap sample.

- What is the probability that x_1 is not the 1st element in the bootstrap sample? What is the probability that it is not the 2nd?
- What is the probability that x_1 is not in the bootstrap sample at all?
- What is the probability that x_2 is not in the bootstrap sample?

Now suppose we have n unique observations $\{x_1, \ldots, x_n\}$ and we create a bootstrap sample.

- What is the probability that x_1 is not in the bootstrap sample?
- What is the probability that an arbitrary observation x_i is not in the bootstrap sample?

What happens to the probability that an arbitrary observation is not in the sample, as n goes to infinity?

Bootstrap Sample Computations Solutions

Suppose we have a sample with 4 unique observations $\{x_1, x_2, x_3, x_4\}$ and create a bootstrap sample.

- What is the probability that x_1 is not the 1st element in the bootstrap sample? What is the probability that it is not the 2nd? $1 - \frac{1}{4}$
- What is the probability that x_1 is not in the bootstrap sample at all? $(1 - \frac{1}{4})^4$
- What is the probability that x_2 is not in the bootstrap sample? $(1 - \frac{1}{4})^4$

Now suppose we have n unique observations $\{x_1, \ldots, x_n\}$ and we create a bootstrap sample.

- What is the probability that x_1 is not in the bootstrap sample? $(1 - \frac{1}{n})^n$
- What is the probability that an arbitrary observation x_i is not in the bootstrap sample? $(1 - \frac{1}{n})^n$

What happens to the probability that an arbitrary observation is not in the sample, as n goes to infinity? $\lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n = e^{-1}$

Bagging

Suppose we only have one training set, but still want to build an ensemble of regression tree models. How can we do it?

Bagging

Suppose we only have one training set, but still want to build an ensemble of regression tree models. How can we do it?

- Bagging (**B**ootstrap **agg**regation) was one of the earliest ensemble techniques

Bagging

Suppose we only have one training set, but still want to build an ensemble of regression tree models. How can we do it?

- Bagging (**B**ootstrap **agg**regation) was one of the earliest ensemble techniques

To create a bagged model, create many bootstrap samples from the original training set, and fit a decision tree to each. Average the resulting predictions.

Bagging

Suppose we only have one training set, but still want to build an ensemble of regression tree models. How can we do it?

- Bagging (**B**ootstrap **agg**regation) was one of the earliest ensemble techniques

To create a bagged model, create many bootstrap samples from the original training set, and fit a decision tree to each. Average the resulting predictions.

Why?

Bagging

Suppose we only have one training set, but still want to build an ensemble of regression tree models. How can we do it?

- Bagging (**B**ootstrap **agg**regation) was one of the earliest ensemble techniques

To create a bagged model, create many bootstrap samples from the original training set, and fit a decision tree to each. Average the resulting predictions.

Why?

- Recall that decision trees tend to have high variance. But averaging the results of independent (or weakly dependent) variables decreases variance
    - Think about the Central Limit Theorem:
    - For large n, $\bar{X}$ is approximately Normal, with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$

Bagging

Suppose we only have one training set, but still want to build an ensemble of regression tree models. How can we do it?

- Bagging (**B**ootstrap **agg**regation) was one of the earliest ensemble techniques

To create a bagged model, create many bootstrap samples from the original training set, and fit a decision tree to each. Average the resulting predictions.

Why?

- Recall that decision trees tend to have high variance. But averaging the results of independent (or weakly dependent) variables decreases variance
    - Think about the Central Limit Theorem:
    - For large n, $\bar{X}$ is approximately Normal, with mean μ and standard deviation $\frac{\sigma}{\sqrt{n}}$
- Unlike a single tree model, we do not prune (we instead control variance by averaging)

Test Error for Bagged Models

- Previously, we showed that an individual observation has probability $e^{-1} \approx 0.368$ of not appearing in a bootstrap sample.

Test Error for Bagged Models

- Previously, we showed that an individual observation has probability $e^{-1} \approx 0.368$ of not appearing in a bootstrap sample.
- For each bootstrap, approximately 1/3 of observations are not included (called *out-of-bag* observations)

Test Error for Bagged Models

- Previously, we showed that an individual observation has probability $e^{-1} \approx 0.368$ of not appearing in a bootstrap sample.
- For each bootstrap, approximately $1/3$ of observations are not included (called *out-of-bag* observations)
- The out-of-bag observations can be used as a natural validation set for the bootstrap model.

Test Error for Bagged Models

- Previously, we showed that an individual observation has probability $e^{-1} \approx 0.368$ of not appearing in a bootstrap sample.
- For each bootstrap, approximately $1/3$ of observations are not included (called *out-of-bag* observations)
- The out-of-bag observations can be used as a natural validation set for the bootstrap model.
- We get an overall estimate of test MSE for the bagged model by averaging the MSE of each bootstrap model on its out-of-bag observations

A Bag of Trees

We return to the pdxTrees data set, this time expanding both our data set size and number of predictors:

```
names(my_pdxTrees)
```

```
## [1] "DBH"                      "Condition"
## [3] "Tree_Height"              "Crown_Width_NS"
## [5] "Crown_Width_EW"           "Crown_Base_Height"
## [7] "Functional_Type"          "Mature_Size"
## [9] "Carbon_Sequestration_lb"
```

```
dim(my_pdxTrees)
```

```
## [1] 3015    9
```

```
set.seed(1)
library(rsample)
my_pdxTrees_split <- initial_split(my_pdxTrees )
my_pdxTrees_train <- training(my_pdxTrees_split)
my_pdxTrees_test <- testing(my_pdxTrees_split)
```

A Bag of Trees

We return to the `pdxTrees` data set, this time expanding both our data set size and number of predictors:

```r
names(my_pdxTrees)
```

```
## [1] "DBH"                        "Condition"
## [3] "Tree_Height"                "Crown_Width_NS"
## [5] "Crown_Width_EW"             "Crown_Base_Height"
## [7] "Functional_Type"            "Mature_Size"
## [9] "Carbon_Sequestration_lb"
```

```r
dim(my_pdxTrees)
```

```
## [1] 3015    9
```

```r
set.seed(1)
library(rsample)
my_pdxTrees_split <- initial_split(my_pdxTrees )
my_pdxTrees_train <- training(my_pdxTrees_split)
my_pdxTrees_test <- testing(my_pdxTrees_split)
```

- Can we improve on our previous model predicting `Carbon_Sequestration_lb`, now using more data and more predictors?

Bagged pdXTrees

- Let’s get a few bootstrap samples using `rsample`:

Bagged pdXTrees

- Let's get a few bootstrap samples using `rsample`:

```r
library(rsample)
set.seed(1115)
pdx_bootstrap <- bootstraps(my_pdxTrees_train, times = 4)
```

Bagged pdXTrees

- Let's get a few bootstrap samples using `rsample`:

```
library(rsample)
set.seed(1115)
pdx_bootstrap <- bootstraps(my_pdxTrees_train, times = 4)
```

- And now build trees on each:

Bagged pdXTrees

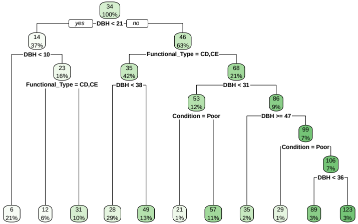
- Let's get a few bootstrap samples using `rsample`:

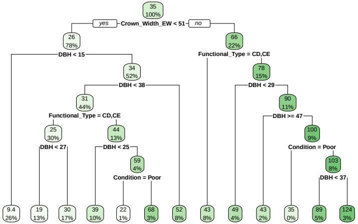
```
library(rsample)
set.seed(1115)
pdx_bootstrap <- bootstraps(my_pdxTrees_train, times = 4)
```

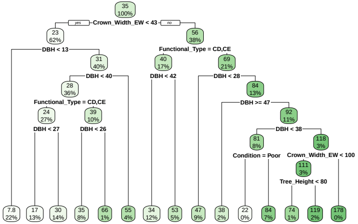
- And now build trees on each:

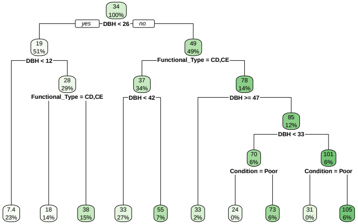
```
library(rpart)
get_tree <- function(split){
  bootstrap_sample <- analysis(split)
  model <- rpart(Carbon_Sequestration_lb ~., data = bootstrap_sample)
}
pdx_bootstrap$model <- map(pdx_bootstrap$splits, get_tree)
```

A few trees


34 100%
yes DBH < 21 no
14 37%
46 63%
DBH < 10
Functional_Type = CD,CE
23 16%
35 42%
68 21%
Functional_Type = CD,CE
DBH < 38
DBH < 31
53 12%
86 9%
Condition = Poor
DBH >= 47
99 7%
Condition = Poor
106 7%
DBH < 36
6 21%
12 6%
31 10%
28 29%
49 13%
21 1%
57 11%
35 2%
29 1%
89 3%
123 3%


35 100%
yes Crown_Width_EW < 51 no
26 78%
66 22%
DBH < 15
Functional_Type = CD,CE
34 52%
78 15%
DBH < 38
DBH < 29
31 44%
90 11%
Functional_Type = CD,CE
DBH >= 47
25 30%
44 13%
100 9%
DBH < 27
DBH < 25
59 4%
Condition = Poor
Condition = Poor
103 8%
DBH < 37
9.4 26%
19 13%
30 17%
39 10%
22 1%
68 3%
52 8%
43 8%
49 4%
43 2%
35 0%
89 5%
124 3%


35 100%
yes Crown_Width_EW < 43 no
23 62%
56 38%
DBH < 13
Functional_Type = CD,CE
31 40%
40 17%
69 21%
DBH < 40
DBH < 42
DBH < 28
28 36%
84 13%
Functional_Type = CD,CE
DBH >= 47
24 27%
39 10%
92 11%
DBH < 27
DBH < 26
DBH < 38
81 8%
118 3%
Condition = Poor
Crown_Width_EW < 100
111 3%
Tree_Height < 80
7.8 22%
17 13%
30 14%
35 8%
66 1%
55 4%
34 12%
53 5%
47 9%
38 2%
22 0%
84 7%
74 1%
119 2%
178 0%


34 100%
yes DBH < 26 no
19 51%
49 49%
DBH < 12
Functional_Type = CD,CE
28 29%
Functional_Type = CD,CE
37 34%
78 14%
DBH < 42
DBH >= 47
85 12%
DBH < 33
70 6%
101 6%
Condition = Poor
Condition = Poor
7.4 23%
18 14%
38 15%
33 27%
55 7%
33 2%
24 0%
73 6%
31 0%
105 6%

Performance

- Let’s get predictions for each bootstrap:

Performance

- Let's get predictions for each bootstrap:

```
get_predictions <- function(model){
  predictions <- predict(model, my_pdxTrees_test)
  data.frame(obs = my_pdxTrees_test$Carbon_Sequestration_lb, preds = predictions)
}
pdx_bootstrap$predictions <- map(pdx_bootstrap$model, get_predictions)
```

Performance

- Let's get predictions for each bootstrap:

```
get_predictions <- function(model){
  predictions <- predict(model, my_pdxTrees_test)
  data.frame(obs = my_pdxTrees_test$Carbon_Sequestration_lb, preds = predictions)
}
pdx_bootstrap$predictions <- map(pdx_bootstrap$model, get_predictions)
```

- And calculate `rmse` on each using `yardstick`

Performance

- Let's get predictions for each bootstrap:

```
get_predictions <- function(model){
  predictions <- predict(model, my_pdxTrees_test)
  data.frame(obs = my_pdxTrees_test$Carbon_Sequestration_lb, preds = predictions)
}
pdx_bootstrap$predictions <- map(pdx_bootstrap$model, get_predictions)
```

- And calculate `rmse` on each using `yardstick`

```
library(yardstick)
results <- map_dfr(pdx_bootstrap$predictions, rmse, obs, preds)
results
```

```
## # A tibble: 4 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        14.0
## 2 rmse    standard        14.3
## 3 rmse    standard        14.3
## 4 rmse    standard        13.1
```

```
mean(results$.estimate)
```

```
## [1] 13.89024
```

Variation in Model Predictions

- How do individual tree predictions compare?

Variation in Model Predictions

- How do individual tree predictions compare?

```
## # A tibble: 6 x 5
## # Rowwise:
##   tree1 tree2 tree3 tree4 bagged
##   <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1  49.0  30.4  52.2  32.5   41.0
## 2  34.7  38.0  43.3  32.8   37.2
## 3  56.8  84.0  67.6  72.9   70.3
## 4  30.6  46.6  38.8  37.8   38.4
## 5  56.8  65.7  67.6  72.9   65.7
## 6  56.8  84.0  89.1  72.9   75.7
```

Variation in Model Predictions

- How do individual tree predictions compare?

```
## # A tibble: 6 x 5
## # Rowwise:
##   tree1 tree2 tree3 tree4 bagged
##   <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1  49.0  30.4  52.2  32.5   41.0
## 2  34.7  38.0  43.3  32.8   37.2
## 3  56.8  84.0  67.6  72.9   70.3
## 4  30.6  46.6  38.8  37.8   38.4
## 5  56.8  65.7  67.6  72.9   65.7
## 6  56.8  84.0  89.1  72.9   75.7
```

- How does the bagged model RMSE compare to each individual tree's RMSE?

Variation in Model Predictions

- How do individual tree predictions compare?

```
## # A tibble: 6 x 5
## # Rowwise:
##   tree1 tree2 tree3 tree4 bagged
##   <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1  49.0  30.4  52.2  32.5   41.0
## 2  34.7  38.0  43.3  32.8   37.2
## 3  56.8  84.0  67.6  72.9   70.3
## 4  30.6  46.6  38.8  37.8   38.4
## 5  56.8  65.7  67.6  72.9   65.7
## 6  56.8  84.0  89.1  72.9   75.7
```

- How does the bagged model RMSE compare to each individual tree's RMSE?

```
## # A tibble: 5 x 4
##   model  .metric .estimator .estimate
##   <chr>  <chr>   <chr>          <dbl>
## 1 tree 1 rmse    standard        14.0
## 2 tree 2 rmse    standard        14.3
## 3 tree 3 rmse    standard        14.3
## 4 tree 4 rmse    standard        13.1
## 5 bagged rmse    standard        12.3
```

Variation in Model Predictions

- How do individual tree predictions compare?

```
## # A tibble: 6 x 5
## # Rowwise:
##   tree1 tree2 tree3 tree4 bagged
##   <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1  49.0  30.4  52.2  32.5   41.0
## 2  34.7  38.0  43.3  32.8   37.2
## 3  56.8  84.0  67.6  72.9   70.3
## 4  30.6  46.6  38.8  37.8   38.4
## 5  56.8  65.7  67.6  72.9   65.7
## 6  56.8  84.0  89.1  72.9   75.7
```

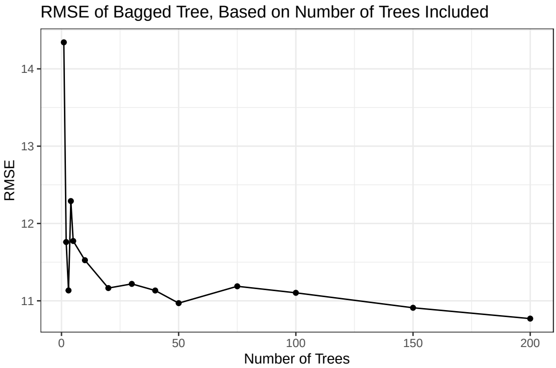
- How does the bagged model RMSE compare to each individual tree's RMSE?

```
## # A tibble: 5 x 4
##   model  .metric .estimator .estimate
##   <chr>  <chr>   <chr>          <dbl>
## 1 tree 1 rmse    standard        14.0
## 2 tree 2 rmse    standard        14.3
## 3 tree 3 rmse    standard        14.3
## 4 tree 4 rmse    standard        13.1
## 5 bagged rmse    standard        12.3
```

- Note that the RMSE for the bagged tree is **NOT** simply the average RMSE. It is significantly *lower*!

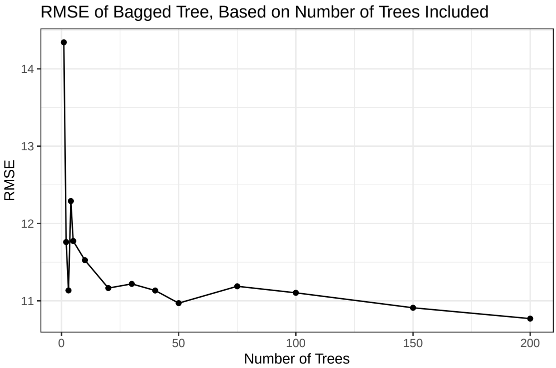
The more trees the merrier?

If 4 trees improved performance over 1, what if we bagged 10 trees? 100?



The more trees the merrier?

If 4 trees improved performance over 1, what if we bagged 10 trees? 100?



- Greatest gains by adding a small number of additional trees
- Moderately small gains thereafter

Section 3

Random Forests

Further Performance Improvements

Suppose we have *m* ensemble models built from the same data set and that it turns out that all *m* models are very similar.

Further Performance Improvements

Suppose we have *m* ensemble models built from the same data set and that it turns out that all *m* models are very similar.

- Do we expect the ensemble model to have high or low variance?

Further Performance Improvements

Suppose we have m ensemble models built from the same data set and that it turns out that all m models are very similar.

- Do we expect the ensemble model to have high or low variance?
  - High variance (since the models are very correlated)

Further Performance Improvements

Suppose we have m ensemble models built from the same data set and that it turns out that all m models are very similar.

- Do we expect the ensemble model to have high or low variance?
    - High variance (since the models are very correlated)
- When bagging trees, if one predictor accounts for large amount of deviation in the response, it will usually be selected as the first split (regardless of the bootstrap sample used)

Further Performance Improvements

Suppose we have *m* ensemble models built from the same data set and that it turns out that all *m* models are very similar.

- Do we expect the ensemble model to have high or low variance?
  - High variance (since the models are very correlated)
- When bagging trees, if one predictor accounts for large amount of deviation in the response, it will usually be selected as the first split (regardless of the bootstrap sample used)
- To artificially increase the variety among trees, we randomly restrict which predictors can be used at each split point.

Further Performance Improvements

Suppose we have *m* ensemble models built from the same data set and that it turns out that all *m* models are very similar.

- Do we expect the ensemble model to have high or low variance?
  - High variance (since the models are very correlated)
- When bagging trees, if one predictor accounts for large amount of deviation in the response, it will usually be selected as the first split (regardless of the bootstrap sample used)
- To artificially increase the variety among trees, we randomly restrict which predictors can be used at each split point.
- Although counterintuitive, this restriction tends to increase accuracy of the ensemble by breaking correlations among the participant trees

Random Forests

To create a random forest:

1. Select the number of models m to build and a number of predictors k to use at each step t
2. Generate a bootstrap sample for each model
3. Build a tree on the bootstrap sample where at each step, a random selection of k of the p predictors can be used (independent of prior predictors selected)
4. Aggregate the models to create an ensemble model.

Random Forests

To create a random forest:

1. Select the number of models m to build and a number of predictors k to use at each step t
2. Generate a bootstrap sample for each model
3. Build a tree on the bootstrap sample where at each step, a random selection of k of the p predictors can be used (independent of prior predictors selected)
4. Aggregate the models to create an ensemble model.

Advantages of the random forest?

Random Forests

To create a random forest:

1. Select the number of models m to build and a number of predictors k to use at each step t
2. Generate a bootstrap sample for each model
3. Build a tree on the bootstrap sample where at each step, a random selection of k of the p predictors can be used (independent of prior predictors selected)
4. Aggregate the models to create an ensemble model.

Advantages of the random forest?

- Individual models are less correlated, so ensemble has lower variance
- Each tree is quicker to build (why?)

Random Forests

To create a random forest:

1. Select the number of models m to build and a number of predictors k to use at each step t
2. Generate a bootstrap sample for each model
3. Build a tree on the bootstrap sample where at each step, a random selection of k of the p predictors can be used (independent of prior predictors selected)
4. Aggregate the models to create an ensemble model.

Advantages of the random forest?

- Individual models are less correlated, so ensemble has lower variance
- Each tree is quicker to build (why?)

Disadvantages?

Random Forests

To create a random forest:

1. Select the number of models m to build and a number of predictors k to use at each step t
2. Generate a bootstrap sample for each model
3. Build a tree on the bootstrap sample where at each step, a random selection of k of the p predictors can be used (independent of prior predictors selected)
4. Aggregate the models to create an ensemble model.

Advantages of the random forest?

- Individual models are less correlated, so ensemble has lower variance
- Each tree is quicker to build (why?)

Disadvantages?

- Difficult to interpret
- Theoretically properties less well-studied (possible MAP project!)

Hand-made Random Forests

I have a data set with categorical response Y with two, with 3 quantitative predictors and with 50 observations.

- Our goal is to build, as a class, a random forest for predicting Y.
- Each table will be tasked with building (by hand) a single decision tree for predicting Y.
- Each table will be randomly assigned 2 of the 3 predictors, and will have a bootstrap sample of the 50 observations.
- Each table will be given a scatterplot showing the relationship between their 2 predictors and the response, on their bootstrap sample.
- Each table should work together to decide where to make cuts in the scatterplot to create a decision tree with between 3 and 6 leaves (group's choice)
- I will give each group the same 10 test points to classify. And as a class, we will average the predictions to create a random forest prediction.

Section 4

Bagging and Random Forests in R

Random Forest in R

- To create both bagged trees and random forests, we use the `randomForest` function in the `randomForest` package in R:

```
library(randomForest)
rfmodel <- randomForest(Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train)
rfmodel
```

```
##
## Call:
##  randomForest(formula = Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 112.2864
##                     % Var explained: 85.74
```

Modifications

We can control how many trees are generated with `ntree` and the number of predictors at each split with `mtry`

Modifications

We can control how many trees are generated with `ntree` and the number of predictors at each split with `mtry`

- By default, `randomForest` uses $p/3$ predictors for regression and $\sqrt{p}$ predictors for classification

Modifications

We can control how many trees are generated with `ntree` and the number of predictors at each split with `mtry`

- By default, `randomForest` uses $p/3$ predictors for regression and $\sqrt{p}$ predictors for classification

```
set.seed(1)
rfmodel2 <- randomForest(Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train,
                         ntree = 10, mtry = 5)
rfmodel2
```

```
##
## Call:
##  randomForest(formula = Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train,      ntree = 1
##                Type of random forest: regression
##                      Number of trees: 10
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 106.4475
##                     % Var explained: 86.48
```

Modifications

We can control how many trees are generated with `ntree` and the number of predictors at each split with `mtry`

- By default, `randomForest` uses $p/3$ predictors for regression and $\sqrt{p}$ predictors for classification

```
set.seed(1)
rfmodel2 <- randomForest(Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train,
                         ntree = 10, mtry = 5)
rfmodel2
```

```
##
## Call:
##  randomForest(formula = Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train,      ntree = 1
##                Type of random forest: regression
##                      Number of trees: 10
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 106.4475
##                     % Var explained: 86.48
```

How can we create a bagged model using the `randomForest` function?

Modifications

We can control how many trees are generated with `ntree` and the number of predictors at each split with `mtry`

- By default, `randomForest` uses $p/3$ predictors for regression and $\sqrt{p}$ predictors for classification

```
set.seed(1)
rfmodel2 <- randomForest(Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train,
                         ntree = 10, mtry = 5)
rfmodel2
```

```
##
## Call:
##  randomForest(formula = Carbon_Sequestration_lb ~ ., data = my_pdxTrees_train,      ntree = 1
##                Type of random forest: regression
##                      Number of trees: 10
## No. of variables tried at each split: 5
##
##           Mean of squared residuals: 106.4475
##                     % Var explained: 86.48
```

How can we create a bagged model using the `randomForest` function?

- Set `mtry=` `p`, where `p` is the total number predictors available

Making predictions

- So you have your `randomForest` model. How do you make predictions?

```
my_preds<- predict(rfmodel, my_pdxTrees_test)
results <- data.frame(obs =  my_pdxTrees_test$Carbon_Sequestration_lb, preds = my_preds)

results %>% head()
```

```
##     obs    preds
## 1  39.0 38.85781
## 2 110.2 66.09302
## 3  61.2 75.53011
## 4  34.0 33.41863
## 5  75.4 51.02538
## 6  96.1 82.35864
```

Variable Importance

Bagging and Random Forests increase prediction accuracy by reducing variance of the model.

Variable Importance

Bagging and Random Forests increase prediction accuracy by reducing variance of the model.

- But the cost comes in interpretability We no longer have a single decision tree to follow to reach our prediction.

Variable Importance

Bagging and Random Forests increase prediction accuracy by reducing variance of the model.

- But the cost comes in interpretability We no longer have a single decision tree to follow to reach our prediction.
- How can we determine which predictors are most influential?

Variable Importance

Bagging and Random Forests increase prediction accuracy by reducing variance of the model.

- But the cost comes in interpretability We no longer have a single decision tree to follow to reach our prediction.
- How can we determine which predictors are most influential?

One possibility is to record the total amount of RSS/Purity that is decreased due to splits of the given predictor, averaged across all trees in the random forest.

Importance in R

```
importance(rfmodel)
```

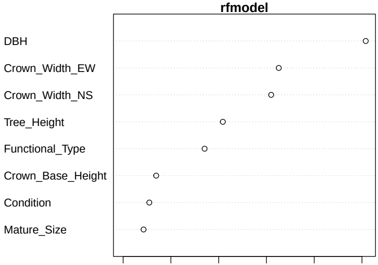
```
##                   IncNodePurity
## DBH                   507862.05
## Condition              54821.17
## Tree_Height           208750.74
## Crown_Width_NS        309930.48
## Crown_Width_EW        325846.81
## Crown_Base_Height      69137.26
## Functional_Type       170538.17
## Mature_Size            42785.73
```

Importance in R

```r
importance(rfmodel)
```

```
##                    IncNodePurity
## DBH                    507862.05
## Condition               54821.17
## Tree_Height            208750.74
## Crown_Width_NS         309930.48
## Crown_Width_EW         325846.81
## Crown_Base_Height       69137.26
## Functional_Type        170538.17
## Mature_Size             42785.73
```

```r
varImpPlot(rfmodel)
```

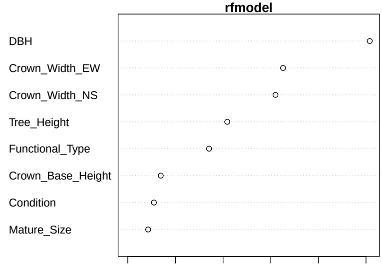


Importance in R

```r
importance(rfmodel)
```

```
##                     IncNodePurity
## DBH                     507862.05
## Condition                54821.17
## Tree_Height             208750.74
## Crown_Width_NS          309930.48
## Crown_Width_EW          325846.81
## Crown_Base_Height        69137.26
## Functional_Type         170538.17
## Mature_Size              42785.73
```

```r
varImpPlot(rfmodel)
```



- For regression trees, node impurity is calculated using RSS.
- For classification trees, node impurity is calculated using Gini Index.

Comparison of Bagged Trees versus Random Forests