Ensemble Models: Bagging

Hunter Glanz

OUTLINE

Foundational Machine Learning

- You've learned about:
  - Traditional Regression
  - Logistic Regression
  - K-Nearest Neighbors
  - Discriminant Analysis
  - Support Vector Machines
  - Tree-Based Methods

Foundational Machine Learning

- You've learned about:
  - Traditional Regression
  - Logistic Regression
  - K-Nearest Neighbors
  - Discriminant Analysis
  - Support Vector Machines
  - Tree-Based Methods

Remember *there's no free lunch*!

Ensemble Learning Strategies

- Ensemble learning refers to algorithms that combine the predictions from two or more models:
  - Let's team up!

Ensemble Learning Strategies

- Ensemble learning refers to algorithms that combine the predictions from two or more models:
  - Let's team up!
  - Near infinite number of ways to do this so we'll talk generally about three broad strategies:
    1. Bagging
    2. Stacking
    3. Boosting

Ensemble Learning Strategies

- Ensemble learning refers to algorithms that combine the predictions from two or more models:
  - Let's team up!
  - Near infinite number of ways to do this so we'll talk generally about three broad strategies:
    1. Bagging
    2. Stacking
    3. Boosting

Today we will focus on **bagging**

Motivation

- Decision trees suffer from *high variance*
  - $\Longrightarrow$ The fit could be quite different depending on the data used

Motivation

- Decision trees suffer from *high variance*
    - $\Longrightarrow$ The fit could be quite different depending on the data used
- *Bagging* (or *bootstrap aggregation*) is a general-purpose procedure for reducing the variance of a statistical learning method

Motivation

- Decision trees suffer from *high variance*
  - $\Longrightarrow$ The fit could be quite different depending on the data used
- *Bagging* (or *bootstrap aggregation*) is a general-purpose procedure for reducing the variance of a statistical learning method
  - Particularly useful and frequently used with decision trees

Motivation

- Decision trees suffer from *high variance*
  - $\implies$ The fit could be quite different depending on the data used
- *Bagging* (or *bootstrap aggregation*) is a general-purpose procedure for reducing the variance of a statistical learning method
  - Particularly useful and frequently used with decision trees

Think back to the sampling distribution of $\bar{X}$. What was the variance of $\bar{X}$?

1. 1
2. σ^2
3. σ^2/n

Motivation

- Decision trees suffer from *high variance*
  - $\implies$ The fit could be quite different depending on the data used
- *Bagging* (or *bootstrap aggregation*) is a general-purpose procedure for reducing the variance of a statistical learning method
  - Particularly useful and frequently used with decision trees

Think back to the sampling distribution of $\bar{X}$. What was the variance of $\bar{X}$?

1. 1
2. σ^2
3. σ^2/n

- $\implies$ Averaging a set of observations reduces variance

The General Idea

- Take many training sets from the population
  - $Z^1, \ldots, Z^B$

The General Idea

- Take many training sets from the population
  - $Z^1, \ldots, Z^B$
- Build a separate prediction model using each training set
  - $\hat{f}^1, \ldots, \hat{f}^B$

The General Idea

- Take many training sets from the population
  - $Z^1, \ldots, Z^B$
- Build a separate prediction model using each training set
  - $\hat{f}^1, \ldots, \hat{f}^B$
- Average the resulting predictions!
  - $\implies$ Single low-variance statistical learning model

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

The Reality

- We don't have multiple training sets, so we bootstrap
  - $Z^{*1}, \ldots, Z^{*B}$

The Reality

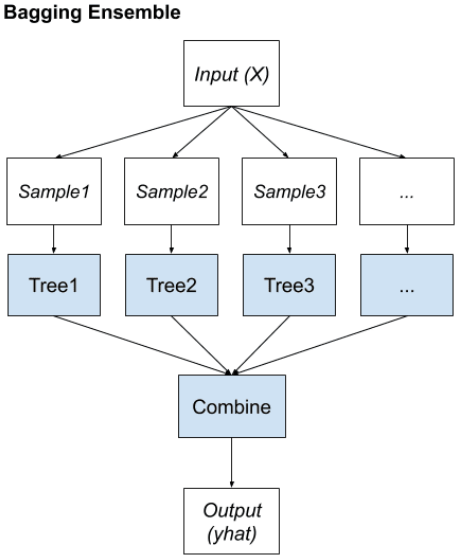
- We don't have multiple training sets, so we bootstrap
  - $Z^{*1}, \ldots, Z^{*B}$
- Build a separate prediction model using each training set
  - $\hat{f}^{*1}, \ldots, \hat{f}^{*B}$

The Reality

- We don't have multiple training sets, so we bootstrap
  - $Z^{*1}, \ldots, Z^{*B}$
- Build a separate prediction model using each training set
  - $\hat{f}^{*1}, \ldots, \hat{f}^{*B}$
- Average the resulting predictions!
  - $\Longrightarrow$ Single low-variance statistical learning model

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

Bagging Visual



When to Bag

- Bagging can improve predictions for many methods
- Bagging is particularly useful for decisions trees because...

When to Bag

- Bagging can improve predictions for many methods
- Bagging is particularly useful for decisions trees because...
  - Large trees have high variance, but low bias

When to Bag

- Bagging can improve predictions for many methods
- Bagging is particularly useful for decisions trees because...
  - Large trees have high variance, but low bias
  - Construct B trees (grown deep with no pruning) based on the B bootstrapped training sets, and then average the predictions

When to Bag

- Bagging can improve predictions for many methods
- Bagging is particularly useful for decisions trees because...
  - Large trees have high variance, but low bias
  - Construct B trees (grown deep with no pruning) based on the B bootstrapped training sets, and then average the predictions
- **If we're performing classification instead, how should we obtain a prediction since we can't take an average?**

Out-of-Bag Error

- *Out-of-Bag* Error
  - On average, each bagged tree makes use of around 2/3 of the observations
  - The remaining 1/3 are referred to as the *out-of-bag* (OOB) observations
  - Make predictions for each observation using trees for which the observation was OOB
  - Compute overall OOB MSE using these predictions

Out-of-Bag Error

- *Out-of-Bag* Error
  - On average, each bagged tree makes use of around 2/3 of the observations
  - The remaining 1/3 are referred to as the *out-of-bag* (OOB) observations
  - Make predictions for each observation using trees for which the observation was OOB
  - Compute overall OOB MSE using these predictions
- For sufficiently large B, OOB error is virtually equivalent to leave-one-out cross-validation error

Out-of-Bag Error

- *Out-of-Bag* Error
  - On average, each bagged tree makes use of around 2/3 of the observations
  - The remaining 1/3 are referred to as the *out-of-bag* (OOB) observations
  - Make predictions for each observation using trees for which the observation was OOB
  - Compute overall OOB MSE using these predictions
- For sufficiently large B, OOB error is virtually equivalent to leave-one-out cross-validation error
- Much easier to compute than using cross-validation when dealing with a very large data set

Bagged Model Assessment

- Number of trees B is not a critical parameter

Bagged Model Assessment

- Number of trees B is not a critical parameter
  - A very large value of B will not lead to overfitting
  - We just want the error estimate to be stable

Bagged Model Assessment

- Number of trees B is not a critical parameter
  - A very large value of B will not lead to overfitting
  - We just want the error estimate to be stable
- Variable Importance
  - Bagging improves prediction accuracy at the expense of interpretability, since we have many trees now
  - For each variable, average the RSS (or Gini index) reduction across all B trees
  - **[Larger]/[Smaller]** values indicate an important predictor

Motivation

- Suppose there is one very strong predictor in the data set, along with many other moderately strong predictors

Motivation

- Suppose there is one very strong predictor in the data set, along with many other moderately strong predictors
- Most or all of the bagged trees will use this strong predictor in the top split

Motivation

- Suppose there is one very strong predictor in the data set, along with many other moderately strong predictors
- Most or all of the bagged trees will use this strong predictor in the top split
  - $\Longrightarrow$ All of the bagged trees will look similar to each other and produce highly correlated predictions

Motivation

- Suppose there is one very strong predictor in the data set, along with many other moderately strong predictors
- Most or all of the bagged trees will use this strong predictor in the top split
  - $\Longrightarrow$ All of the bagged trees will look similar to each other and produce highly correlated predictions
- Averaging highly correlated quantities does not reduce variance as much as averaging uncorrelated quantities

How Random Forests Address Correlation in the Trees

- Build decision trees just like we would in *bagging*

How Random Forests Address Correlation in the Trees

- Build decision trees just like we would in *bagging*
- However, each time a split in a tree is considered, *a random sample of m predictors* is chosen as split candidates

How Random Forests Address Correlation in the Trees

- Build decision trees just like we would in *bagging*
- However, each time a split in a tree is considered, *a random sample of m predictors* is chosen as split candidates
  - A fresh set of m predictors is taken at each split

How Random Forests Address Correlation in the Trees

- Build decision trees just like we would in *bagging*
- However, each time a split in a tree is considered, *a random sample of m predictors* is chosen as split candidates
  - A fresh set of m predictors is taken at each split
- **What happens if $m = p$?**

How Random Forests Address Correlation in the Trees

- Build decision trees just like we would in *bagging*
- However, each time a split in a tree is considered, *a random sample of m predictors* is chosen as split candidates
  - A fresh set of m predictors is taken at each split
- **What happens if $m = p$?**
- Typically we choose $m \approx \sqrt{p}$

Customizing Random Forests

- As with bagging, random forests will not overfit for large numbers of trees (large B)
  - $\Longrightarrow$ We just want B sufficiently large

Customizing Random Forests

- As with bagging, random forests will not overfit for large numbers of trees (large B)
  - $\implies$ We just want B sufficiently large
- Different values of m could affect the performance: