INFO 6105
Data Science Engineering Methods and Tools

Lecture 9
Bagging and Random Forests

Ebrahim Nasrabadi
nasrabadi@northeastern.edu

College of Engineering
Northeastern University

Fall 2019

Advantages and Disadvantages of Trees

- Very easy to interpret

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn
- Handle qualitative predictors without the need to create dummy variables

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables
- Unfortunately, a single decision tree does not perform well

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables
- Unfortunately, a single decision tree does not perform well
- Additionally, trees can be very non-robust.

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables
- Unfortunately, a single decision tree does not perform well
- Additionally, trees can be very non-robust.

Advantages and Disadvantages of Trees

- Very easy to interpret
- Easy to visualize (especially if they are small) and fast to learn
- Handle qualitative predictors without the need to create dummy variables
- Invariant to monotone transformation of predictor variables
- Unfortunately, a single decision tree does not perform well
- Additionally, trees can be very non-robust.

Today's lecture: Improve the performance of trees substantially by aggregating many decision trees, using methods like bagging, random forests, and boosting

The Bias-Variance Analysis

- We want to predict y given $\mathbf{x}$

The Bias-Variance Analysis

- We want to predict y given $\mathbf{x}$
- Assume that the true function is $y = f(\mathbf{x}) + \epsilon$

- We want to predict y given $\mathbf{x}$
- Assume that the true function is $y = f(\mathbf{x}) + \epsilon$
  - where ϵ is normally distributed with zero mean and standard deviation σ.

The Bias-Variance Analysis

- We want to predict y given $\mathbf{x}$
- Assume that the true function is $y = f(\mathbf{x}) + \epsilon$
  - where ϵ is normally distributed with zero mean and standard deviation σ.
- Given a set of training examples,

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$$

we fit a hypothesis $h_D(\mathbf{x})$ to the data to minimize the squared error

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - h_D(\mathbf{x}_i))^2$$

The Bias-Variance Analysis

- In general, we do not really care how well the method works on the training data (training error)

The Bias-Variance Analysis

- In general, we do not really care how well the method works on the training data (training error)
- We are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data (test error)

The Bias-Variance Analysis

- In general, we do not really care how well the method works on the training data (training error)
- We are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data (test error)
- Now, given a new data point $\mathbf{x}_0$ with observed value

$$y_0 = f(\mathbf{x}_0) + \epsilon,$$

we would like to understand the expected prediction error

$$\mathrm{E}_{D,\epsilon}\left[(f(\mathbf{x}_0) + \epsilon - h_D(x_0))^2\right]$$

The Bias-Variance Analysis

- Note that

$$\mathrm{E}_{D,\epsilon}\left[(f(\mathbf{x}_0) + \epsilon - h_D(x_0))^2\right]$$

calculates the average test MSE that we would obtain if we repeatedly estimated h using a large number of training sets, and tested each at $\mathbf{x}_0$.

- Note that
$$\mathrm{E}_{D,\epsilon}\left[\left(f(\mathbf{x}_0)+\epsilon-h_D(x_0)\right)^2\right]$$
calculates the average test MSE that we would obtain if we repeatedly estimated h using a large number of training sets, and tested each at $\mathbf{x}_0$.
- The overall expected test MSE can be computed by averaging
$$\mathrm{E}_{D,\epsilon}\left[\left(f(\mathbf{x}_0)+\epsilon-h_D(x_0)\right)^2\right]$$
over all possible data points $\vec{x}_0$ in the test set.

Notation

- Let's simplify

$$\mathrm{E}_{D,\epsilon}\left[\left(f(\mathbf{x}_0)+\epsilon-h_D(x_0)\right)^2\right] \quad \text{as} \quad \mathrm{E}_{D,\epsilon}\left[(f-\hat{y})^2\right]$$

  where
  - f is used to denote $f(\mathbf{x}_0)+\epsilon$
  - $\hat{y}$ is used to denote $h_D(x_0)$

Notation

- Let's simplify

$$\mathrm{E}_{D,\epsilon}\left[\left(f(\mathbf{x}_0)+\epsilon-h_D(x_0)\right)^2\right] \quad \text{as} \quad \mathrm{E}_{D,\epsilon}\left[(f-\hat{y})^2\right]$$

  where
  - f is used to denote $f(\mathbf{x}_0)+\epsilon$
  - $\hat{y}$ is used to denote $h_D(x_0)$
- Recall that
  - $f(\mathbf{x})$ is the true value
  - ϵ is the noise
  - D is the training data set
  - $h_D(\mathbf{x}_0)$ is the hypothesis learned from D

Notation

- Let's simplify

$$\mathrm{E}_{D,\epsilon}\left[\left(f(\mathbf{x}_0)+\epsilon-h_D(x_0)\right)^2\right] \quad \text{as} \quad \mathrm{E}_{D,\epsilon}\left[(f-\hat{y})^2\right]$$

  where
  - f is used to denote $f(\mathbf{x}_0)+\epsilon$
  - $\hat{y}$ is used to denote $h_D(x_0)$
- Recall that
  - $f(\mathbf{x})$ is the true value
  - ϵ is the noise
  - D is the training data set
  - $h_D(\mathbf{x}_0)$ is the hypothesis learned from D
- Let h denote long-term expectation of prediction on $\mathbf{x}_0$ averaged over many data sets D:

$$h = E_D[h_D(\mathbf{x}_0)]$$

Bias – Variance decomposition of error

- Note that

$$
\begin{aligned}
&\mathrm{E}_{D,\epsilon}\left[(f-\hat{y})^2\right] \\
&=E\left[((f-h)-(\hat{y}-h))^2\right] \\
&=E\left[(f-h)^2+(h-\hat{y})^2+2(f-h)(h-\hat{y})\right] \\
&=E\left[(f-h)^2+(h-\hat{y})^2+2(fh-f\hat{y}-h^2+h\hat{y})\right] \\
&=E\left[(f-h)^2\right]+E\left[(h-\hat{y})^2\right]+2E[fh-f\hat{y}-h^2+h\hat{y}] \\
&=E\left[(f-h)^2\right]+E\left[(h-\hat{y})^2\right]
\end{aligned}
$$

- The last equality is derived from the fact that

$$
\begin{aligned}
E[fh-f\hat{y}-h^2+h\hat{y}] &= fh-fE[\hat{y}]-h^2+hE[\hat{y}] \\
&= fh-fh-h^2+h^2 \\
&= 0
\end{aligned}
$$

Bias – Variance decomposition of error

- The expected test MSE can be decomposed into BIAS^2 and VARIANCE as

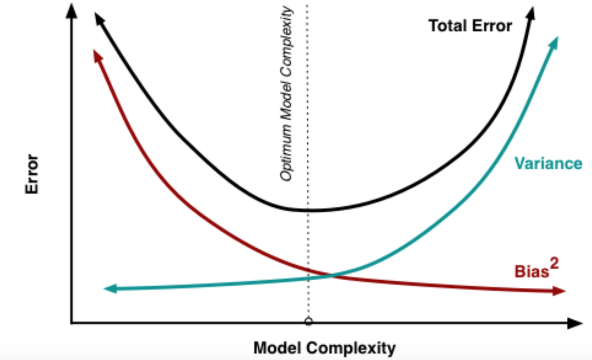
$$\begin{aligned} \mathrm{E}_{D,\epsilon}\left[(f-\hat{y})^2\right] &= E\left[(f-h)^2\right] + E\left[(h-\hat{y})^2\right] \\ &= \text{BIAS}^2 + \text{VARIANCE} \end{aligned}$$

- Note that
  - BIAS^2 gives the squared difference between the true value and our "long-term" expectation for what the learner will do if we averaged over many datasets D
  - VARIANCE gives the squared difference between our long-term expectation for the model performance, $E_D[h_D(x)]$, and what we expect in a representative run on a dataset D (as $\hat{y}$)

Bias-variance trade-off

- The relationship between bias, variance, and test error is referred to as the bias-variance trade-off.
- Good test set performance requires low variance as well as low squared bias.
- This is referred to as a trade-off because it is easy to obtain a method with
  - extremely low bias but high variance or
  - very low variance but high bias
- The challenge lies in finding a method for which both the variance and the squared bias are low.

Bias-variance trade-off



Tradeoff between bias and variance

- Simple Models: High Bias, Low Variance
- Complex Models: Low Bias, High Variance

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a machine learning method

Bagging – Motivation

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a machine learning method
- Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean Z of the observations is given by σ^2/n.

Bagging – Motivation

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a machine learning method
- Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean Z of the observations is given by σ^2/n.
- In other words, averaging a set of observations reduces variance.

Bagging – Motivation

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a machine learning method
- Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean Z of the observations is given by σ^2/n.
- In other words, averaging a set of observations reduces variance.
- In practice, we have only ONE data set, say D.

Bagging – Motivation

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a machine learning method
- Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean Z of the observations is given by σ^2/n.
- In other words, averaging a set of observations reduces variance.
- In practice, we have only ONE data set, say D.
- Instead, we can bootstrap by repeatedly sampling observations from the original data set.

Bagging – Motivation

- Bootstrap aggregation, or bagging, is a general-purpose procedure for reducing the variance of a machine learning method
- Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean Z of the observations is given by σ^2/n.
- In other words, averaging a set of observations reduces variance.
- In practice, we have only ONE data set, say D.
- Instead, we can bootstrap by repeatedly sampling observations from the original data set.
- The sampling is performed with replacement, which means that the same observation can occur more than once in the bootstrap data set.

Bagging Regression Trees

- Generate B (hundreds) different bootstrap training data sets, say $D_1, \ldots, D_B$

Bagging Regression Trees

- Generate B (hundreds) different bootstrap training data sets, say $D_1, \ldots, D_B$
- Apply the learning algorithm to each bootstrap sample D_b to get $h_b(\mathbf{x})$, the prediction at a point $\mathbf{x}$.

Bagging Regression Trees

- Generate B (hundreds) different bootstrap training data sets, say $D_1, \ldots, D_B$
- Apply the learning algorithm to each bootstrap sample D_b to get $h_b(\mathbf{x})$, the prediction at a point $\mathbf{x}$.
- Average all the predictions to obtain

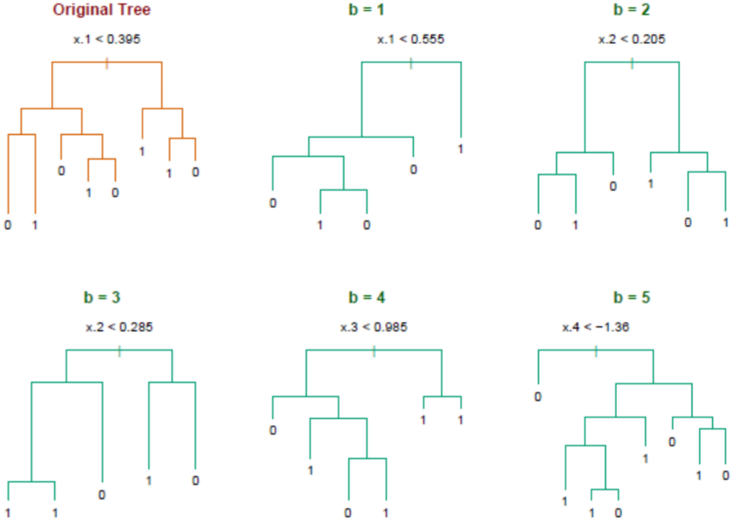
$$h_{bag}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} h_b(\mathbf{x})$$

This is called bagging.

Bagging Classification Trees

- The above prescription applied to regression trees
- For classification trees: for each test observation,
  - record the class predicted by each of the B trees,
  - then take a majority vote: the overall prediction is the most commonly occurring class among the B predictions.
- If we are interested in the posterior probabilities, we can rather average the class proportions in the terminal nodes.

Bagging Classification Trees



Hastie et al.,"The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer (2009)

Bagging – issues

- Each tree is identically distributed (i.d.) (not independent and identically distributed (i.i.d.))
  - the expectation of the average of B such trees is the same as the expectation of any one of them
  - the bias of bagged trees is the same as that of the individual trees

Bagging – issues

- An average of B i.i.d. random variables, each with variance σ^2, has variance: σ^2

Bagging – issues

- An average of B i.i.d. random variables, each with variance σ^2, has variance: σ^2
- If i.d. (identical but not independent) and pair correlation ρ is present, then the variance is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

Bagging – issues

- An average of B i.i.d. random variables, each with variance σ^2, has variance: σ^2
- If i.d. (identical but not independent) and pair correlation ρ is present, then the variance is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

- As B increases the second term disappears but the first term remains

Bagging – issues

Question: Why does bagging generate correlated trees?

Bagging – issues

Question: Why does bagging generate correlated trees?

- Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors.

Bagging – issues

Question: Why does bagging generate correlated trees?

- Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors.
- Then all bagged trees will select the strong predictor at the top of the tree and therefore all trees will look similar.

Bagging – issues

Question: Why does bagging generate correlated trees?

- Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors.
- Then all bagged trees will select the strong predictor at the top of the tree and therefore all trees will look similar.

Bagging – issues

Question: Why does bagging generate correlated trees?

- Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors.
- Then all bagged trees will select the strong predictor at the top of the tree and therefore all trees will look similar.

Question: How to avoid this?

Bagging – issues

Question: Why does bagging generate correlated trees?

- Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors.
- Then all bagged trees will select the strong predictor at the top of the tree and therefore all trees will look similar.

Question: How to avoid this?
Solution: We select randomly a subset of the predictors at each split

Random Forests

- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.

Random Forests

- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.

Random Forests

- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of m predictors. The split is allowed to use only one of those predictors.

Random Forests

- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of m predictors. The split is allowed to use only one of those predictors.
- A fresh selection of predictors is taken at each split

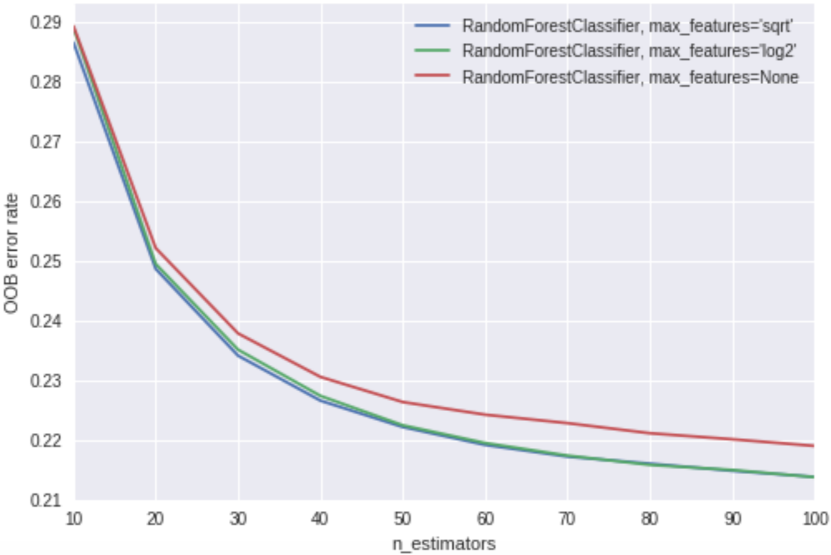
Random Forests

- Random forests provide an improvement over bagged trees by way of a small tweak that decorrelates the trees.
- As in bagging, we build a number of decision trees on bootstrapped training samples.
- But when building these decision trees, each time a split in a tree is considered, a random selection of m predictors is chosen as split candidates from the full set of m predictors. The split is allowed to use only one of those predictors.
- A fresh selection of predictors is taken at each split
- Typically we choose $\sqrt{m}$ variables for classification trees and $m/3$ variables for regression trees.

Out-of-Bag Error Estimation

- There is a very straightforward way to estimate the test error of a bagged model without the need to perform cross-validation or the validation set approach.
- The key to bagging is that trees are repeatedly fit to bootstrapped subsets of the observations.
- One can show that on average, each bagged tree makes use of around 2/3 of the observations.
- The remaining 1/3 of the observations not used to fit a given bagged tree are referred to as the out-of-bag (OOB) observations.
- We can predict the response for the ith observation using each of the trees in which that observation was OOB.
- This will yield around B/3 predictions for the ith observation, which we average.

Out-of-Bag Error Estimation



Feature Importance Measures

- Reduces overfitting (variance)

Feature Importance Measures

- Reduces overfitting (variance)
- Good performance and very popular

Feature Importance Measures

- Reduces overfitting (variance)
- Good performance and very popular
- Easy to parallelize

Feature Importance Measures

- Reduces overfitting (variance)
- Good performance and very popular
- Easy to parallelize
- Unfortunately, difficult to interpret the resulting model. Bagging improves prediction accuracy at the expense of interpretability.

Feature Importance Measures

- Reduces overfitting (variance)
- Good performance and very popular
- Easy to parallelize
- Unfortunately, difficult to interpret the resulting model. Bagging improves prediction accuracy at the expense of interpretability.

Feature Importance Measures

- Reduces overfitting (variance)
- Good performance and very popular
- Easy to parallelize
- Unfortunately, difficult to interpret the resulting model. Bagging improves prediction accuracy at the expense of interpretability.

Feature Importance: Calculate the total amount that the RSS or Gini index is decreased due to splits over a given predictor, averaged over all B trees.

Feature Importance Measures

- We can obtain an overall summary of the importance of each predictor using the RSS (for bagging regression trees) or the Gini index (for bagging classification trees).
- In the case of bagging regression trees, we can record the total amount that the RSS is decreased due to splits over a given predictor, averaged over all B trees.
- A large value indicates an important predictor.
- Similarly, in the context of bagging classification trees, we can add up the total amount that the Gini index is decreased by splits over a given predictor, averaged over all B trees.

Feature Importance Measures

```
Feature ranking:
1. Credit_Score (0.205297)
2. Current_Loan_Amount (0.139297)
3. Maximum_Open_Credit (0.089153)
4. Current_Credit_Balance (0.086238)
5. Monthly_Debt (0.085836)
6. Years_of_Credit_History (0.081426)
7. Annual_Income (0.078193)
8. Number_of_Open_Accounts (0.056693)
9. Months_since_last_delinquent (0.049870)
10. Years_in_current_job (0.042165)
```