# BIOS 835: Bagging and Random Forests

Alexander McLain

October 17, 2023

# Bagging

- Bagging and Boosting are machine learning methods that can be used to improve a model.
- Bagging or *bootstrap aggregation* is a technique for reducing the variance of an estimated prediction function.
- Bagging is particularly effective for predictions that have low bias and high variance.
- We'll discuss bagging and boosting thoroughly in our next class as part of ensemble learning. Today, we'll focus on bagging CART.

# Definition

- Let $\mathbf{Z} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ denote the training data use to obtain an estimate $\hat{f}(x)$.
- Let $\mathbf{Z}^{*b}$ be a bootstrap sample from $\mathbf{Z}$, for $b = 1, 2, \ldots, B$.
- For each bootstrap sample we get an estimate $\hat{f}^{*b}(x)$
- The bagged estimate is defined by

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

which can be thought of as $\mathrm{E}_{\hat{\mathcal{P}}} \hat{f}^{*}(x)$ where $\hat{\mathcal{P}}$ is the empirical distribution function.

# Bagging Trees

- Recall that classification trees are classifiers that have high variance and low bias.
- Suppose our tree produces a classifier $\hat{G}(x)$ for a $K$-class response with

$$\hat{G}(x) = \arg\max_k \hat{f}(x)$$

  where $\hat{f}(x)$ has a single one and $K-1$ zeroes.
- Then the bagged estimate $\hat{f}_{\text{bag}}(x)$ is a $K$-vector $[p_1(x), p_2(x), \ldots p_K(x)]$, with $p_k(x)$ equal to the proportion of trees predicting class $k$ at $x$.
- The bagged classifier selects the class with the most "votes" from the $B$ trees

$$\hat{G}_{\text{bag}}(x) = \arg\max_k \hat{f}_{\text{bag}}(x)$$

# Bagging Notes

- The $p_k(x)$ should not be treated as probabilities of membership.
- For classification, we can understand the bagging effect in terms of a consensus of independent *weak learners*.
- Suppose we have $B$ independent classifiers for a two-sample problem, each with error $1 - \epsilon < 0.5$.
- Let $S_1(x)$ be the sum of our weak classifiers.
  - What's the distribution of $S_1(x)$?
  - What's the probability $S_1(x) > B/2$ as $B \to \infty$?
- This concept has been popularized outside of statistics as the "Wisdom of Crowds"

# Wisdom of crowds

Sir Francis Galton's Ox Experiment (1906):

- Sir Francis Galton, a British polymath and cousin of Charles Darwin, was intrigued by the idea of average judgment and wanted to test its validity. He found his opportunity at a country fair in Plymouth, where an event was being held in which attendees could guess the weight of an ox on display.
- About 800 people, both experts (butchers and farmers) and non-experts alike, took part in the contest, writing down their guesses on tickets.
- After the event, Galton collected and analyzed all the tickets. He wanted to prove that the average guess of the crowd would be way off the mark.
- To his astonishment, the mean of all the guesses (1,197 pounds) was incredibly close to the actual weight of the ox, which was 1,198 pounds!

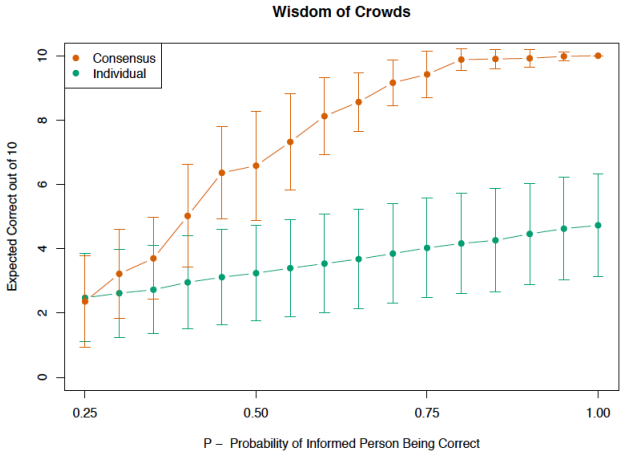


Figure: Simulated academy awards voting. 50 members vote in 10 categories, each with 4 nominations. From page 287 in the online version of ESL.

# Random Forests

- *Random forests* (Breiman, 2001) is a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them.
- *Random forests* are very simple to train and use.
- Bagging is useful when we have many noisy but unbiased classifiers.
- CART are very noisy and unbiased (if they are grown deep enough).

# Random Forests

- If variables are i.d. (not i.i.d.) with correlation $\rho$ then the variance of their average is

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

- Which can be made as small as necessary if $\rho = 0$.
- The idea in random forests: **is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the variance too much.**

# Random Forests Algorithm

1. For $b = 1, 2, \ldots, B$
   - 1.1 Draw a bootstrap sample $\boldsymbol{Z}^*$ of size $N$ from the training data.
   - 1.2 Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.
     - 1.2.1 Select $m$ variables at random from the $p$ variables.
     - 1.2.2 Pick the best variable/split-point among the $m$.
     - 1.2.3 Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

# Random Forests

- The algorithm results in $\{T_b\}_1^B$ which can be used to predict classes or continuous outcomes.
- **Regression:** $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$
- **Classification:** Let $\hat{C}_b(x)$ be the majority vote for the $b$th tree, then

$$\hat{C}_{\text{rf}}^B(x) = \text{ majority vote } \left\{\hat{C}_b(x)\right\}_1^B$$

# Random Forests Details

- Typically values for $m$ depend on whether regression or classification is being done.
- For classification, the default value for $m$ is $\lfloor \sqrt{p} \rfloor$ and the minimum node size is one (also 1 is not a crazy value).
- For regression, the default value for m is $\lfloor p/3 \rfloor$ and the minimum node size is five.
- In practice the best values for these parameters will depend on the problem, and they should be treated as tuning parameter (but they are the only tuning parameter).

# Out-of-bag Samples

- Note that for each bootstrap sample there are observations ($\sim 37\%$) that are not included in the training of the tree. We call these *out-of-bag* ($\mathrm{OOB}$) samples.
- The $\mathrm{OOB}$ samples can be used to estimate the error.
- The error for observation $z_i = (x_i, y_i)$ is defined as it's prediction error for a random forest averaging only those trees corresponding to bootstrap samples in which $z_i$ did not appear.
- Random forests have built in validation!!

# Variable Importance

- There are two methods to construct variable importance plots can be constructed for random forests.
- The first *Gini* approach is described as follows:
  - At each split in each tree, the improvement in the split-criterion is the importance measure attributed to the splitting variable, and is accumulated over all the trees in the forest separately for each variable.
  - The improvement in split-criterion is measured using the *Gini* index.

# Variable Importance

The second variable importance measure uses the OOB samples:

- When the $b$th tree is grown, the OOB samples are passed down the tree, and the prediction accuracy is recorded.
- Then the values for the $j$th variable are randomly permuted in the OOB samples, and the accuracy is again computed.
- The decrease in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable $j$ in the random forest.

# Overview: advantages

- **Accuracy:** Random forests often provide higher accuracy in comparison to single decision trees.
- **Overfitting:** The model is less prone to overfitting than individual decision trees, thanks to averaging out the results.
- **Handling Large Data:** Can efficiently process large datasets and handle datasets with higher dimensionality.
- **Missing Data:** Can handle missing data by using the median to replace continuous variables or computing the weighted average of missing values.
- **Out-of-Bag (OOB) Score:** The OOB sample can be used as a validation set to estimate accuracy, removing the need for a separate validation set.
- **Variable Importance:** Provides insights into feature importance, which can be crucial for domain understanding and feature selection.

# Overview: disadvantages

- **Complexity and size:** The model can be quite large and not as easily exportable and slow for real-time predictions.
- **Interpretability:** Random forests are more complex than single decision trees and are hence less interpretable.
- **Bias:** If the dataset has a dominant class, random forests can be biased towards that class. Balancing the dataset prior can help alleviate this.
- **Parameter Tuning:** They require fewer parameter tunings than some other algorithms, but they still have hyperparameters.
- **Not Always Best:** For data that can be linearly separated, simpler models like logistic regression might perform better.

# Probabilistic Random Forests: overview

- Probabilistic Random Forests (PRFs) build upon the standard approach by also estimating the probability distribution of the output (target) variables, not just the expected value.
- They provide a full distribution of possible outcomes along with the probabilities of these outcomes.
  - For classification, this can be viewed as an extension of the standard practice of estimating class probabilities in a Random Forest, but with a focus on accurately estimating these probabilities and potentially providing more information about their reliability.
  - For regression, instead of predicting a single value for each instance, PRFs predict a probability distribution over possible values, giving a sense of the uncertainty in the predictions.

# Uncertainty Quantification

- The main advantage of PRFs is their ability to quantify the uncertainty in predictions.
- This is crucial in many domains like healthcare, finance, or autonomous vehicles, where making decisions based on predictions requires not only point estimates but also a measure of confidence or reliability.
- This uncertainty estimate can be derived from the variability in predictions across the different trees in the forest.
- If all trees agree closely on a prediction, uncertainty is lower; if there's a wide spread in the predictions, uncertainty is higher.

# Implementation

- Implementing a PRF typically involves modifications to the standard RF algorithm to output probability distributions.
- This can involve changes to:
  - the tree-growing procedure,
  - the way the trees are aggregated, and
  - the loss function used for training.
- Advanced methods might involve fitting a parametric or non-parametric probability distribution to the outputs of the trees for each input instance.

# Considerations

- While PRFs provide valuable information about uncertainty, they can be more complex to implement and interpret than standard RFs.
- The quality of the uncertainty estimates can depend heavily on the data and the specific implementation of the PRF.
- It's important to validate these uncertainty estimates through techniques like cross-validation.