Predictive Modeling

Bagging and Random Forests

Mirko Birbaumer

HSLU T&A

Pros and Cons of Decision Trees

- **Advantages of Decision Trees**:
  - Easily comprehensible and versatile with respect to the predictors (qualitative predictors can be handled without dummy variables)
  - Trees can be visualized nicely

- **Disadvantages of Decision Trees**:
  - Poor predictive accuracy
  - Decision trees have high variance : if we split the training data into two parts at random and fit a decision tree to both halves, the results that we get could be quite different
  - See example `0.1` of the `Random Forests` chapter

Bagging

- **Bagging** allows for reducing the variance of a statistical estimator (e.g. a decision tree)
- Bagging is based on the **aggregation** of a multitude of estimators with high variance by means of averaging and thus reducing the variance
- These estimators are created by **bootstrapping** the training data (in fact, the notion bagging stems from **b**ootstrap **agg**regat**ing**)
- **Random forests** is a further development of bagging being a state-of-the-art classification method for real world problems.

Bootstrapping

- The use of the term **bootstrap** derives from the phrase *to pull oneself up by one's bootstraps*, widely thought to be based on one of the eighteenth century *The surprising adventures of Baron Munchausen* by Rudolphe Erich Raspe

- *The baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, he thaught to pick himself up by his own bootstraps.*

- The bootstrap helps us out of the problem, when we cannot generate new samples from the original population

Bootstrapping

- The bootstrap approach allows us to use a computer to mimic the process of obtaining new data sets, so that we can estimate the variability of our estimate without generating additional samples

- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set **with replacement**

- Each of these **bootstrap data sets** is created by sampling **with replacement** and is the same size as our original data set. As a result, some observations may appear more than once in a given bootstrap data set and some not at all

Bootstrapping

Bootstrapping

Let $x_1, \ldots, x_n$ be (possibly multivariate) realizations of independent and identically distributed random variables $X_1, \ldots, X_n$. Assume further that

$$\widehat{\gamma} = \widehat{\gamma}(x_1, \ldots, x_n)$$

is an estimator of some quantity γ.

1. Choose a (large) number $B \in \mathbb{N}$.

2. For $b = 1, \ldots, B$
   - Draw n samples $\{x_1^*, \ldots, x_n^*\}$ from $\{x_1, \ldots, x_n\}$ with replacement.
   - Compute the estimator $\widehat{\gamma}_b^* = \widehat{\gamma}(x_1^*, \ldots, x_n^*)$

Bootstrapping

Bootstrapping

3. The empirical distribution function $\widehat{F}^*$ of $(\widehat{\gamma}_1^*, \ldots, \widehat{\gamma}_B^*)$ approximates the distribution of $\widehat{\gamma}$. In particular, the standard error of $\widehat{\gamma}$ can be estimated by the bootstrap estimate

$$\mathrm{se}_B(\widehat{\gamma}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\widehat{\gamma}_n^* - \bar{\gamma}^*)^2}, \quad \text{with}$$

$$\bar{\gamma}^* = \frac{1}{B}\sum_{b=1}^{B}\widehat{\gamma}_b^*.$$

Bootstrapping

- **Note** : each individual bootstrap sample $x_1^*, \ldots, x_n^*$ will contain duplicates of values in the original data set and likewise some values will not appear at all, please solve exercises `1` and `2`

- See example `1.1` of the `Random Forests` chapter

Bagging - Bootstrap aggregation

- **Bootstrap aggregation**, or **bagging** : is a general-purpose procedure for reducing the variance of a statistical learning method; it is particularly useful and frequent in the context of decision trees

- Recall that for a sample of independent random variables $Z_1, \ldots, Z_n$ each with variance σ^2, the **variance of the mean** $\bar{Z}$ is only σ^2/n

- By **averaging** we reduce the variance

Bagging - Bootstrap aggregation

- Natural way to decrease the variance of a statistical learning method and hence to **increase the predictive accuracy** would consist of
  - taking many training sets from the population
  - building a **separate prediction model** using each training set
  - **averaging** the resulting predictions

- This procedure is not practical since we do not have access to several training sets

- **Bootstrap** by taking repeated samples from (single) training set

- In other words, we generate B different bootstrapped training sets and for each of these sets we compute a **predictive model** $\widehat{f}_1^*(x), \ldots, \widehat{f}_B^*(x)$

Bagging - Regression Setting

- If we are in the **regression setting** (imagining that each of the $\widehat{f}_b^*$ is a linear regression model) then the bagged model is simply the average

$$\widehat{f}_{\text{bag}}(x) = \frac{1}{B} \sum_{i=1}^{B} \widehat{f}_i^*(x)$$

Bagging - Classification Setting

- For **classification** typically the predicted classes of all B models are recorded and the most commonly occurring class among the B predictions is used.
- This is referred to as **majority vote**
- Please check example `2.1` of the `Random Forests` chapter

Out-of-Bag Error

- 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
- Recall that 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 **two-thirds** of the observations
- The remaining one-third of the observations not used to fit a given bagged tree is 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

Out-of-Bag Error Estimate

Out-of-bag error estimate

Let $(\overrightarrow{x}_1, y_1), \ldots, (\overrightarrow{x}_n, y_n)$ be the training set and $\widehat{f}_1^*, \ldots, \widehat{f}_B^*$ be B bootstrapped classification methods (e.g. decision trees).

1. For $i = 1, \ldots, n$ find all bootstrapped models $\widehat{f}_b^*$ that do **not** use the i-th observation for training. Use these models to make a prediction $\widehat{y}_i^*$ by means of a majority vote

2. The out-of-bag (OOB) error estimate is the classification error

$$\text{Err}^* = \frac{1}{n} \sum_{i=1}^{n} I(y_i \neq \widehat{y}_i^*).$$

Out-of-Bag Error Estimate

- The resulting OOB error is a valid estimate of the test error for the bagged model, since the response for each observation is predicted using only the trees that were not fit using that observation
- It can be shown that with B sufficiently large, OOB error is virtually equivalent to **leave-one-out cross-validation error**
- Particularly convenient when performing bagging on large data sets for which cross-validation would be computationally onerous
- Please check example `2.2` of the `Random Forests` chapter and solve exercises `1` in the exercise sheet

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 sample of m predictors is chosen** as split candidates from the full set of p predictors

Random Forests

- A split in the decision tree is allowed to use **only one of those** m **predictors**

- A fresh sample of m predictors is taken at each split, and typically we choose $m \approx \sqrt{p}$ – that is, the number of predictors considered at each split is approximately equal to the square root of the total number of predictors

- Example `Heart` : 4 out of the 13 for the `Heart` data

- Please check example `3.1` of the `Random Forests` chapter and solve exercise `3` in the exercise sheet

Random Forests

- Building a random forest, at each split in the tree, the algorithm is **not** even allowed to consider a majority of the available predictors

- Clever rationale: Suppose that there is **one very strong predictor** in the data set, along with a number of other moderately strong predictors$\longrightarrow$ in the collection of bagged trees, most or all of the trees will use this strong predictor in the **top split**

- Consequently, all of the bagged trees will look quite similar to each other : predictions from the bagged trees will be highly **correlated**

Random Forests

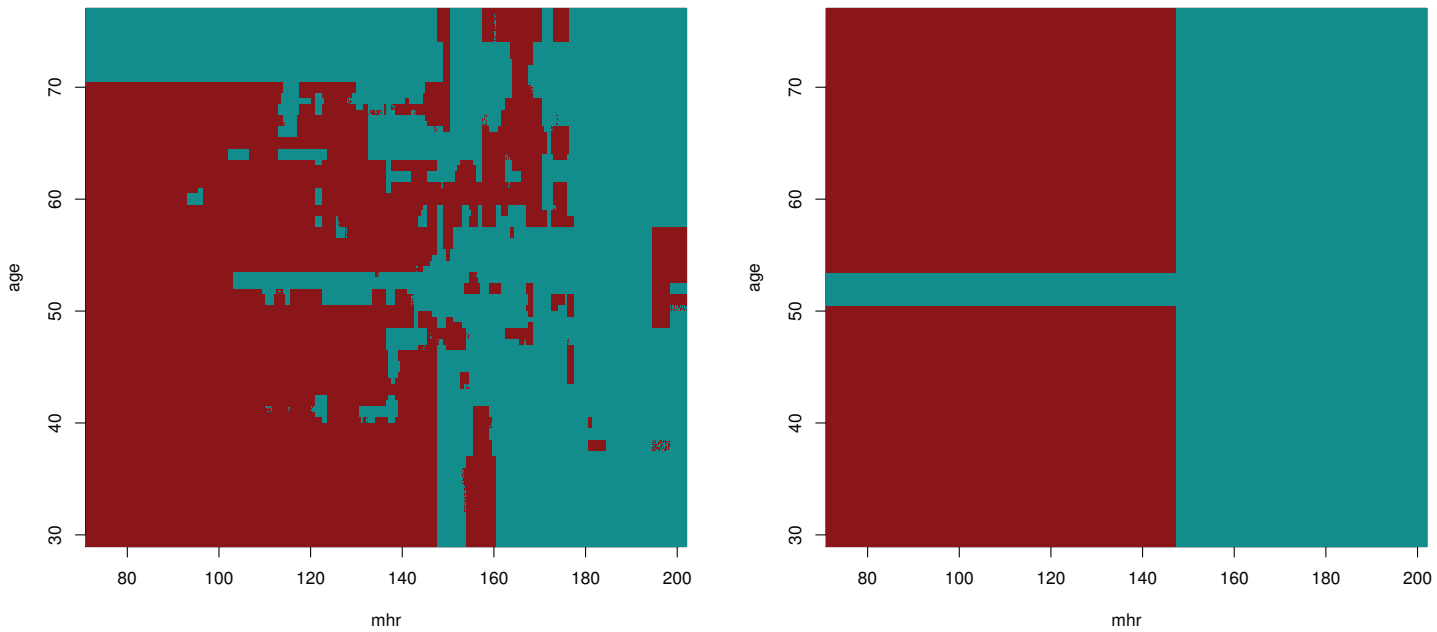
- Unfortunately, **averaging many highly correlated quantities** does **not** lead to as large of a reduction in variance as averaging many uncorrelated quantities

- **Random forests** overcome this problem by forcing each split to consider only a **subset** of the predictors

Random Forests : Variable Importance

- **Bagging** typically results in **improved accuracy** over prediction using a single tree. Unfortunately, it can be difficult to **interpret** the resulting model
- Thus, bagging improves prediction accuracy **at the expense of interpretability**
- One can obtain an overall summary of the **importance of each predictor using the Gini index**
- We can add up the total amount that the **Gini index is decreased** by splits over a given predictor, averaged over all B trees
- This averaged value is termed **variable importance** and tells us for each predictor how large its contribution to the model is ; please check example `3.2` of the `Random Forests` chapter

Decision Trees versus Random Forests



In the left panel the partition resulting from the random forest model is shown, on the right image the binary partition due to an ordinary tree model is depicted. The partition due to the random forest is much more complicated and harder to interpret as the simple binary partition of the tree. Please check example `3.3` of the `Random Forests` chapter