Data splitting, cross validation, and bootstrap

-Applied Analytics-

APM: Chapter 4

Lecturer: Darren Homrighausen, PhD

Preamble:

- Look at over-fitting
- Overview data splitting strategies
- Look into resampling techniques

Overview

All modeling techniques have some built-in flexibility

This flexibility can be characterized by some parameters that can be set by the user

These parameters are called tuning parameters, to disambiguate them from other model parameters

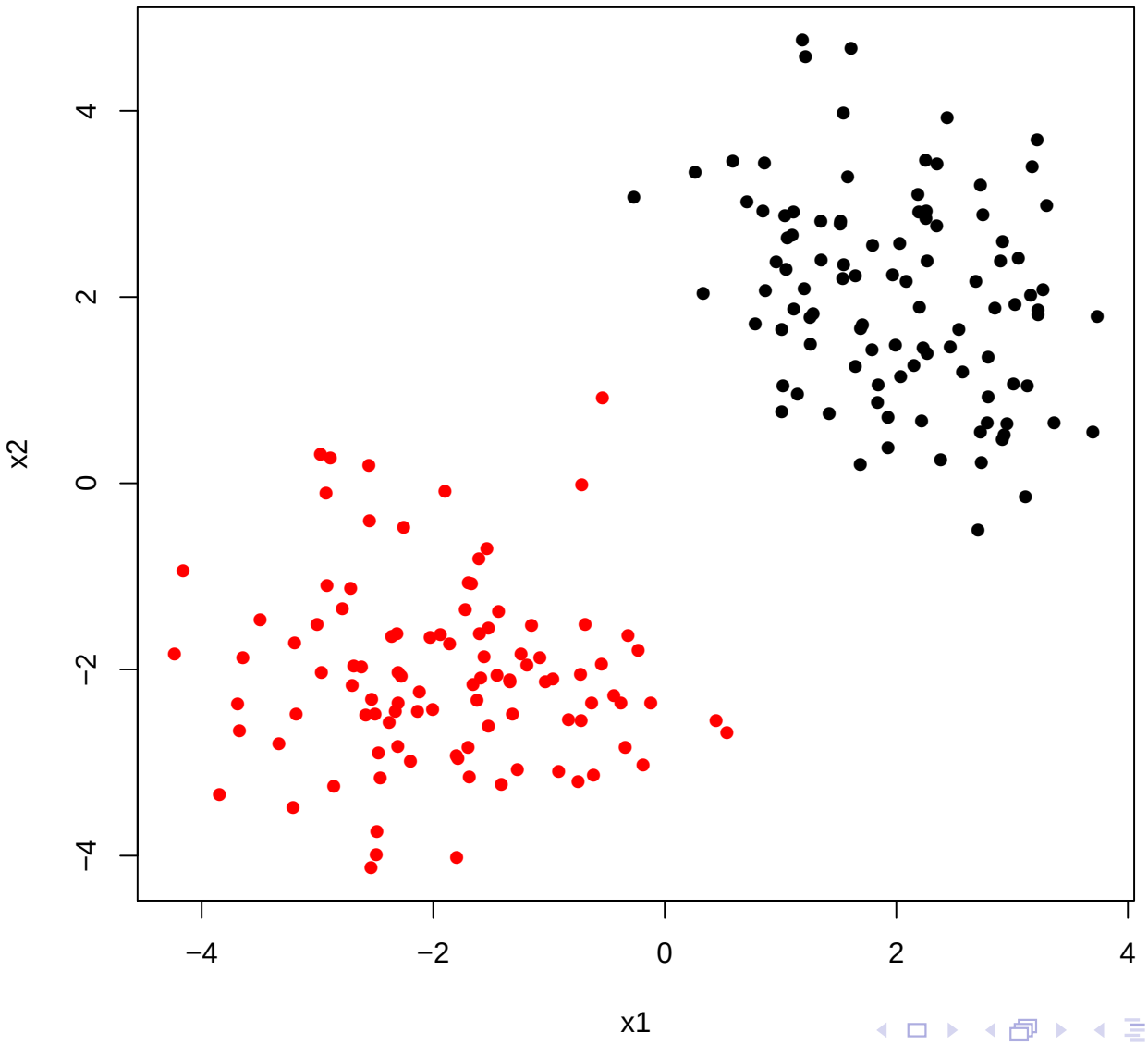
(One example of a model parameter would be the population mean: μ and an example of tuning parameter would be K in K-nearest neighbors)

Much of the applied analytics process (after the data has been processed into a useable data set) involves

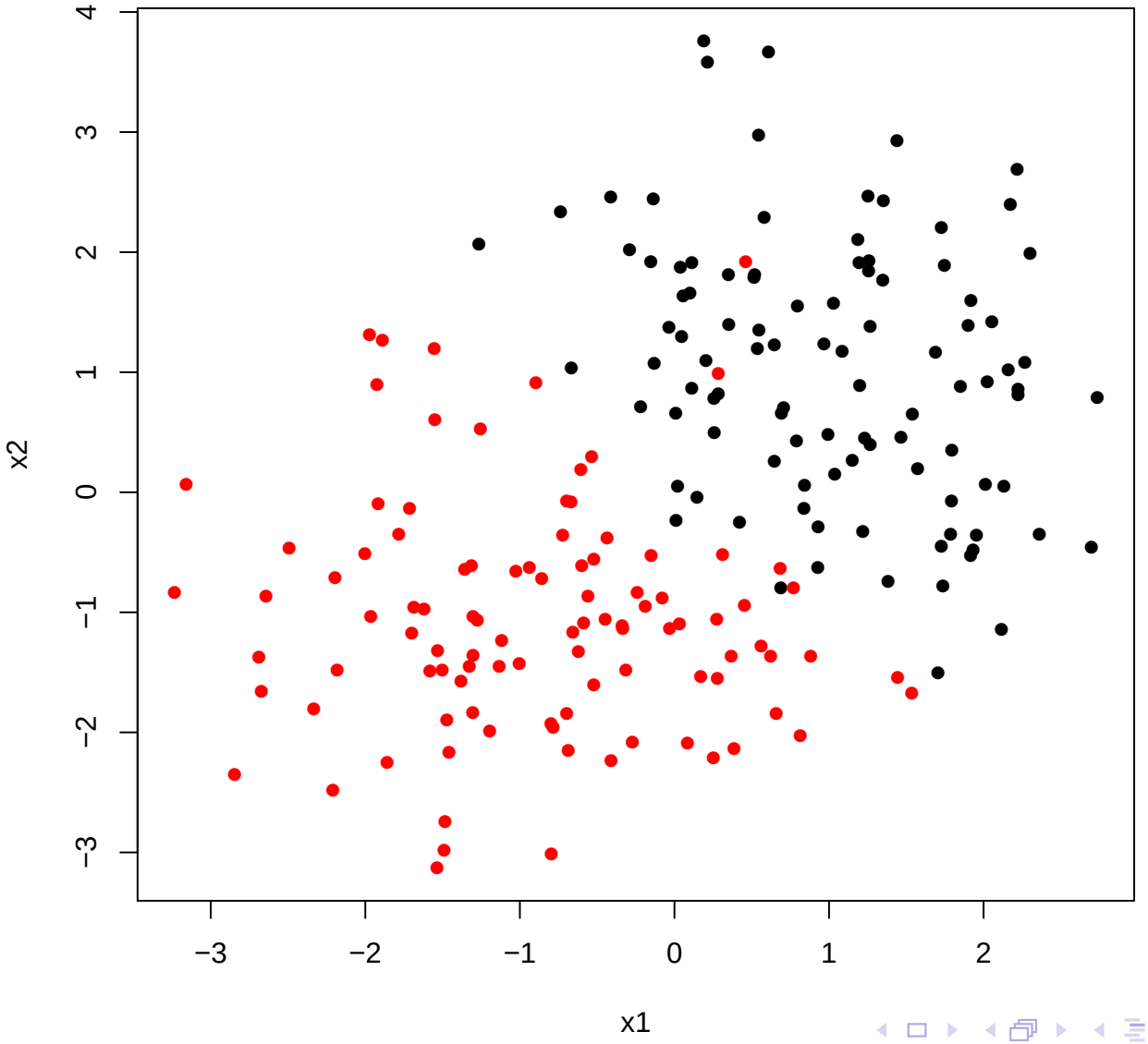
- choosing a modeling technique
- estimating the model parameters
- setting the tuning parameters
- using the final model for its intended purpose

A motivating example

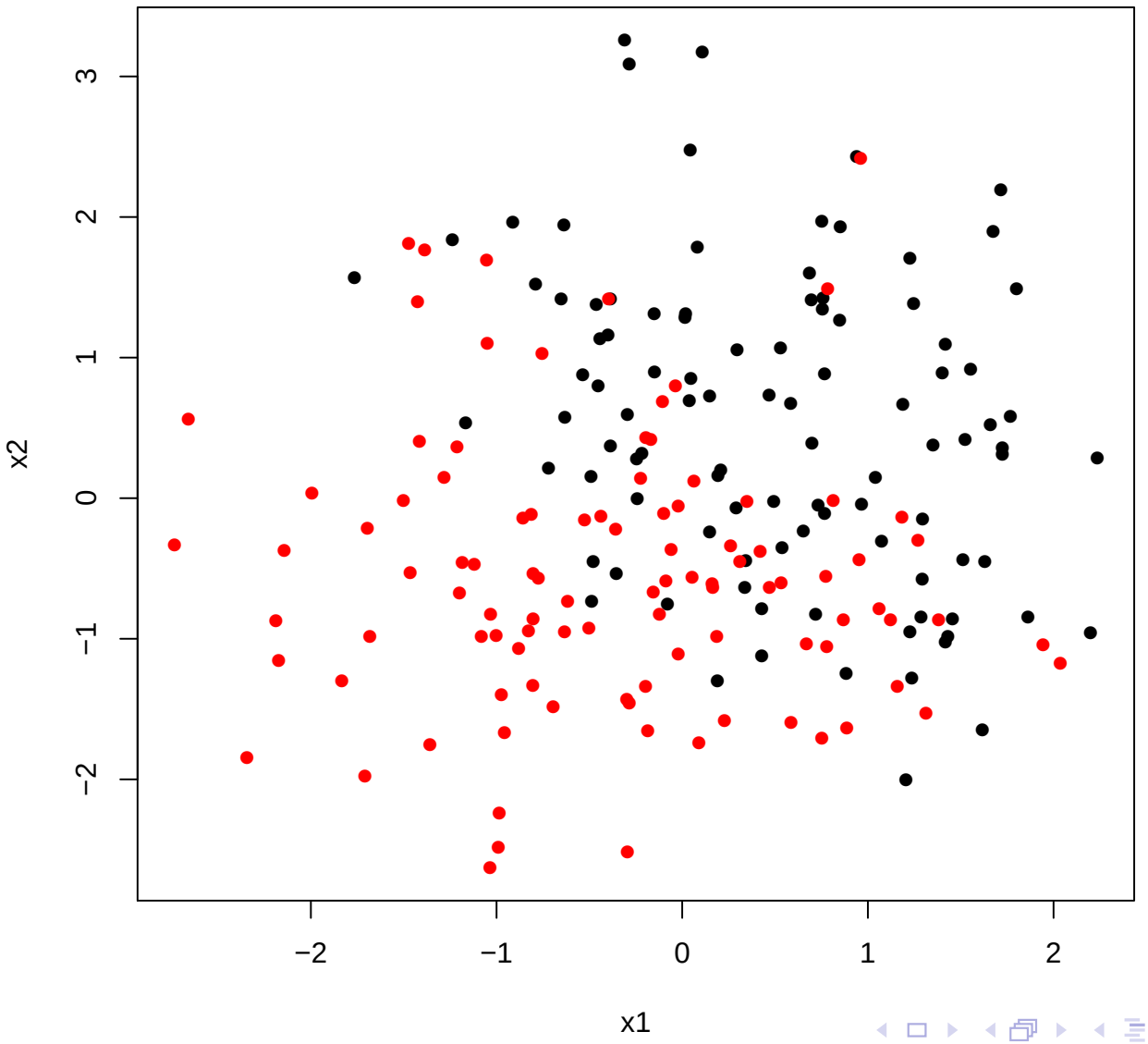
The main issue with setting tuning parameters



The main issue with setting tuning parameters



The main issue with setting tuning parameters



Judging quality

Reminder: The training data is the data set we are using to estimate the model parameters

We additionally need to choose the correct model flexibility so that it has "good" performance

If we judge the quality of the model only on how good the performance is on the training data, then we are likely to over-fit

- The book refers to "how good the performance is on the training data" as the apparent error. It is also generally known as the training error
- It turns out the training error is optimistic in a way that can be made precise

Bottom line: Using the training data to estimate model parameters and then using the training error to judge its quality will lead to sub-standard results

Data splitting

A classic approach

One way to address the problem with the apparent error is via data splitting

Via some mechanism (which we will discuss next) split all your data into 3 non-overlapping subsets:

1. Training: Used to fit (or train) the considered models
2. Validation: Used to choose each models tuning parameter(s)
3. Testing: Used to choose amongst estimated/tuned models

A typical split might be 50%/25%/25%

Data splitting

Some mechanisms by which these subsets can be formed

- Simple random sampling. This means that each observation is randomly assigned with equal probability to one subset
- Stratified random sampling. If the supervisor Y or a feature X is qualitative it makes sense to make each subset mimic the whole data set
- Maximum dissimilarity sampling. Attempts to prevent the subsets from being systematically different from each other. Note that this is non-random

  (In the cited paper Martin et al. (2012) they don't find much evidence to suggest this proposal produces superior results)

Data splitting

Data splitting has some weaknesses

- There needs to be a very large amount of data
- There can be issues with rare features
- The results depend on the splits used to generate the three subsets

What we'd like to know is how sensitive our results are to these original splits..

Resampling techniques provide different ways of doing just that

Resampling

Resampling in general

All of the resampling methods start off by randomly partitioning the data into subsets

(This is just like data splitting at this point)

The difference arises by repeating this process at least two times (and in some cases thousands of times)

Then, we aggregate the results over each resample

(The most common way the results are aggregated is via averaging)

Eventually we will use data splitting and resampling together

(For example, we might do a training/test split and use resampling on the training data)

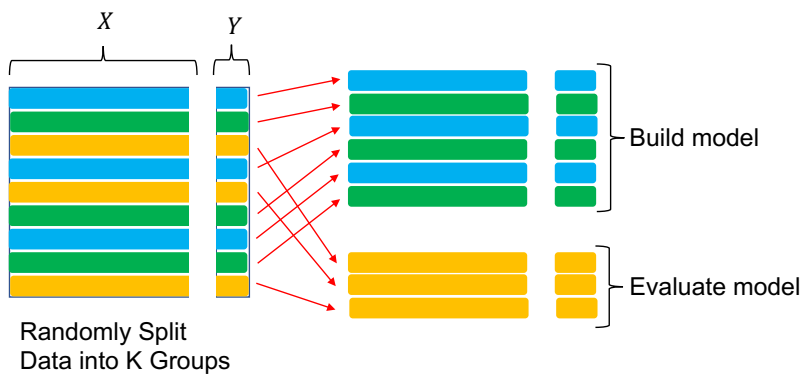
K-fold cross-validation (CV)

The most common method is (K-fold) cross-validation (CV)

The data set is randomly split into K subsets (folds) of (roughly) equal size

(It is still a good idea to do stratified random sampling)

Then each fold takes its turn as a test subset while the other $K-1$ folds get combined together into a training set

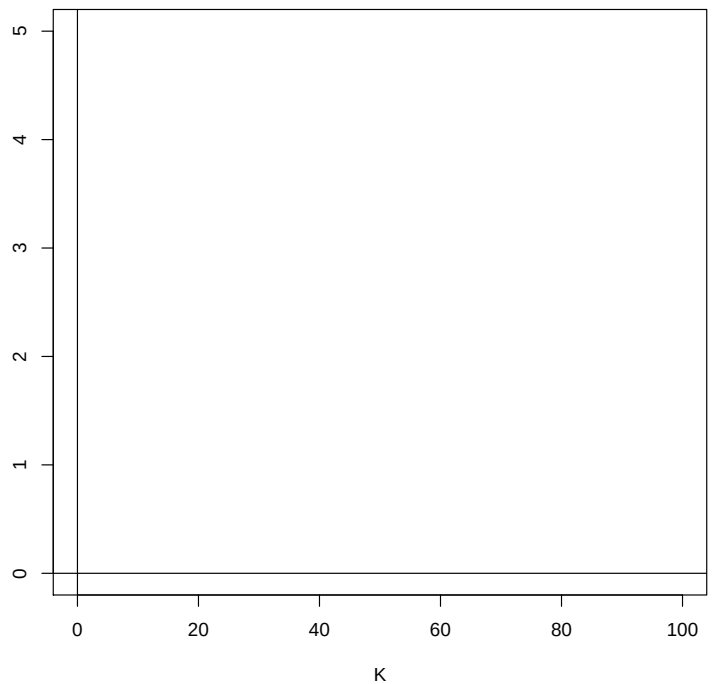


K-fold cross-validation (CV)

It is important to emphasize that CV estimates model performance

As such, we can talk about its bias and variance as an estimator

Suppose our training data has $n = 100$ observations



- Choosing $K = n$ is called leave one out CV (LOOCV)

  (It is more computationally demanding than smaller K but doesn't require randomization)
- Choosing $K = 5$ to $K = 10$ is by far the most common choice

Generalized cross-validation (GCV)

This isn't a cross-validation procedure at all

(The name is from the original motivation rather than descriptive about how it works)

We will return to this later as it has much more in common with AIC/BIC than CV

Monte-Carlo cross-validation

This slight variation on K-fold CV works by choosing a large number B and a fraction of observations to allocate to a training/test split

(Say, something like 0.9 to training and 0.1 to test)

Re-randomize into training/test splits B times and aggregate the results over all B randomization

This would only realistically be considered with very small data sets where there would be substantial concern for getting non-representative folds

Bootstrap

The bootstrap is an incredibly useful tool for quantifying uncertainty

It is especially useful for situations where you want an estimate of the variance or a confidence interval for a...

- ... complicated statistic
- ... simple statistic but with non-normal data and no central limit theorem

Let's take a detour into the bootstrap to discuss it more fully as it is so useful

Bootstrap detour

Bootstrap detour

Suppose we are looking to invest in two financial instruments, x_1 and x_2.

The return on these investments is random, but we still want to allocate our money in a risk minimizing way

(Here, I'm using variance as a proxy for "risk")

That is, for some $a \in (0, 1)$, we want to minimize

$$\mathrm{Var}(ax_1 + (1 - a)x_2)$$

The minimizing a is:

$$a_* = \frac{\sigma_1^2 - \sigma_{12}}{\sigma_2^2 + \sigma_1^2 - 2\sigma_{12}}$$

(Here, σ_{12} is the covariance between x_1 and x_2)

Bootstrap detour

We can estimate a_* via a plug-in estimator

$$\hat{a} = \frac{\hat{\sigma}_1^2 - \hat{\sigma}_{12}^2}{\hat{\sigma}_1^2 + \hat{\sigma}_2^2 - 2\hat{\sigma}_{12}^2}$$

Now that we have an estimator of a_*, it would be nice to have an estimator of its variance

In this case, computing a standard error is difficult

Boostrap detour

The key concept at play is the sampling distribution of an estimator

We generally only have one data set, so we only have one value for an estimator

However, if we consider the data we observed as random, then that means the estimator is random as well

Thus, the estimator has a distribution!

This distribution (or its approximation) is what we are usually after when we want to quantify our uncertainty

Bootstrap detour: Fantasy land

Suppose for a moment that we can simulate a large number of data sets from the same distribution as the data set we observe

Then we could recompute $\hat{a}$ on each of the data sets

Plotting a histogram with the values $\hat{a}_1, \ldots, \hat{a}_{1000}$, we could visualize the sampling distribution of $\hat{a}$

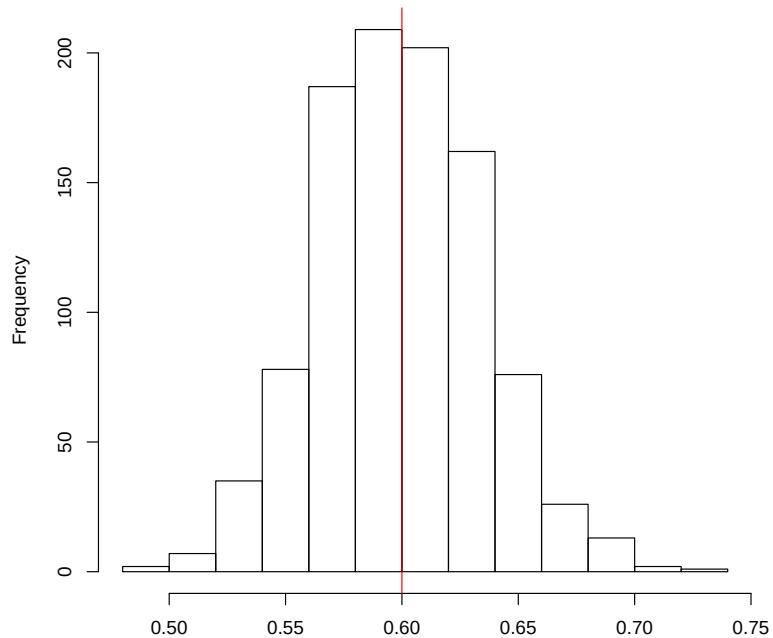
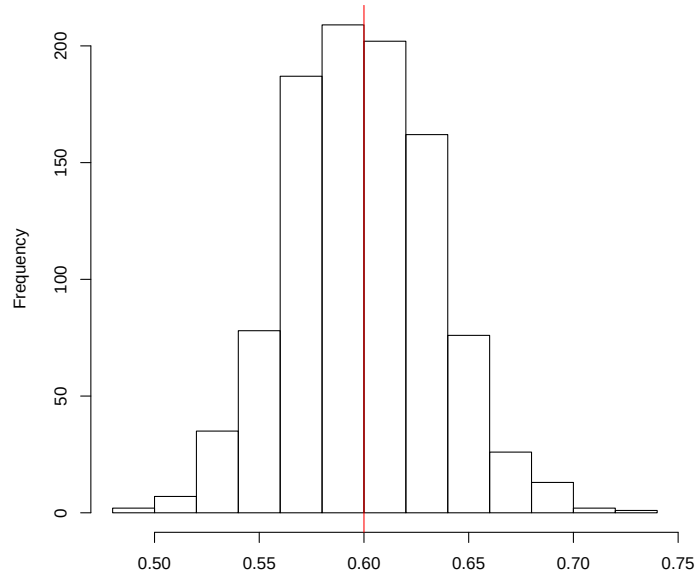


Figure: This is the sampling distribution of $\hat{a}$

Bootstrap detour: Fantasy land



The mean of all of these is:

$$\bar{a} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{a}_r = 0.599,$$

which is very close to 0.6 (red line), and the standard deviation is

$$\sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} (\hat{a}_r - \bar{a})^2} = 0.079$$

Bootstrap detour: Fantasy land

Sometimes the standard deviation is all that is needed

Usually, a confidence interval is desired as well

This can be done via the quantiles of the values $\hat{a}_1, \ldots, \hat{a}_{1000}$

To form a 95% confidence interval, we could

1. sort the $\hat{a}$'s from smallest to largest
2. find the $\hat{a}$ in the list that is the 25^{th} entry, call it L
3. find the $\hat{a}$ in the list that is the 975^{th} entry, call it U
4. form a confidence interval $[L, U]$

In this case, I found a CI of $[0.45, 0.733]$

Bootstrap detour: Reality

In practice, of course, we cannot use this procedure as it relies on being able to draw a large number of new, independent data sets from the same distribution as our data set

This is where the bootstrap comes in.

We instead draw new data sets directly from our observed data set

This sampling is done with replacement, which means that the same data point can be drawn multiple times.

Bootstrap detour: Small example

Suppose we have data $\mathcal{D} = (4.3, 3, 7.2, 6.9, 5.5)$

Then we can draw bootstrap samples, which might look like:

$$\mathcal{D}_1 = (7.2, 4.3, 7.2, 5.5, 6.9)$$
$$\mathcal{D}_2 = (6.9, 4.3, 3.0, 4.3, 6.9)$$
$$\vdots$$
$$\mathcal{D}_B = (4.3, 3.0, 3.0, 5.5, 6.9)$$

It turns out each of these $\mathcal{D}_b$ have similar properties as $\mathcal{D}$

Back to the example, we draw n observations with replacement from $X_i = (x_1, x_2)$

For each bootstrap sample, we recompute all the variance/covariance estimates to form $\hat{a}_b$

Bootstrap detour: Reality

Now, we form the bootstrap mean:

$$\text{mean}_B = \frac{1}{B}\sum_{b=1}^{B} \hat{a}_b$$

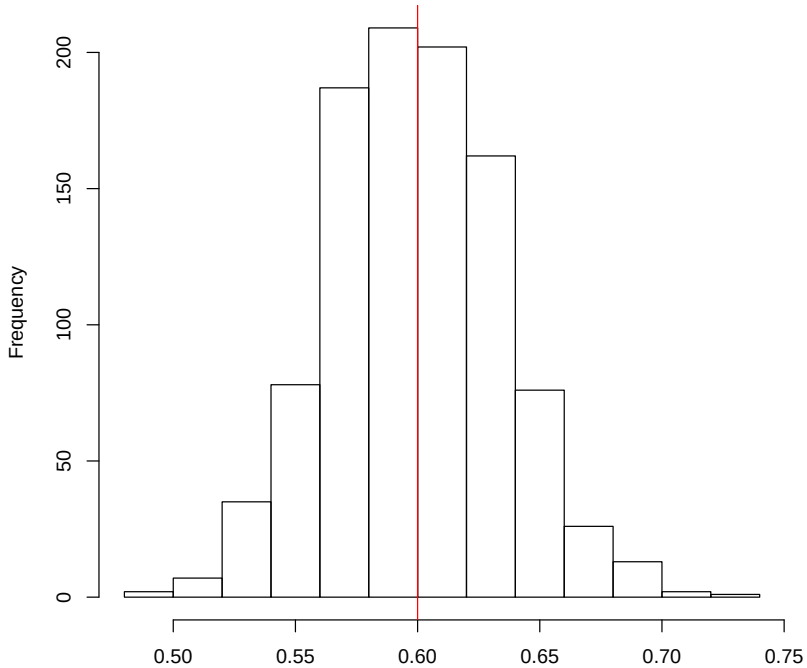
The bootstrap estimator of the standard deviation is:

$$\text{SE}_B = \sqrt{\frac{1}{B}\sum_{b=1}^{B} (\hat{a}_b - \text{mean}_B)^2}$$

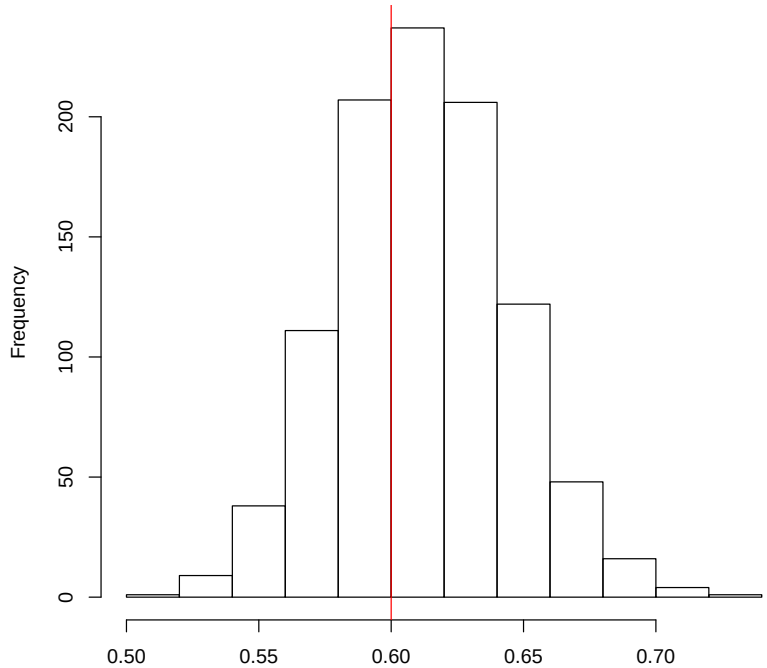
We can still form a $(1-\alpha) * 100\%$ bootstrap confidence interval:

1. sort the $\hat{a}_b$'s from smallest to largest
2. find the $\hat{a}$ in the list that is the $B * \alpha/2^{th}$ entry, call it L
3. find the $\hat{a}$ in the list that is the $B * (1-\alpha/2)^{th}$ entry, call it U
4. form a confidence interval $[L, U]$

Bootstrap detour: Fantasy vs. Reality



Sampling distribution of $\hat{a}$ (impossible to form)



Bootstrap distribution of $\hat{a}$ (possible to form)

End detour

Bootstrap

Back to estimating model performance

We can apply the bootstrap technique by choosing a large number B

Then, for each $b = 1, \ldots, B$, we can

1. Draw n times with replacement from the training data to form $\mathcal{D}_b$
2. Train the model on $\mathcal{D}_b$
3. Estimate the model performance on the training data that isn't in $\mathcal{D}_b$

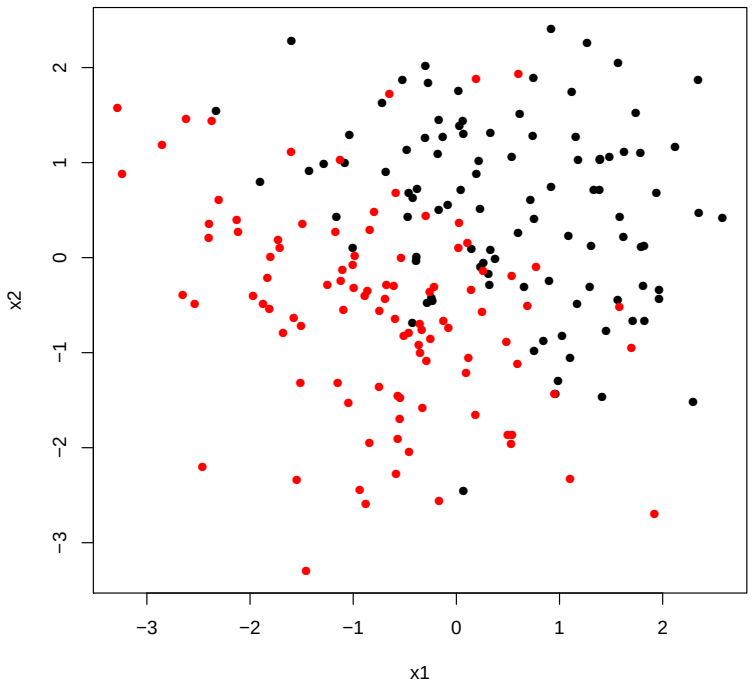
   (Using math notation $\mathcal{D} \setminus \mathcal{D}_b$)

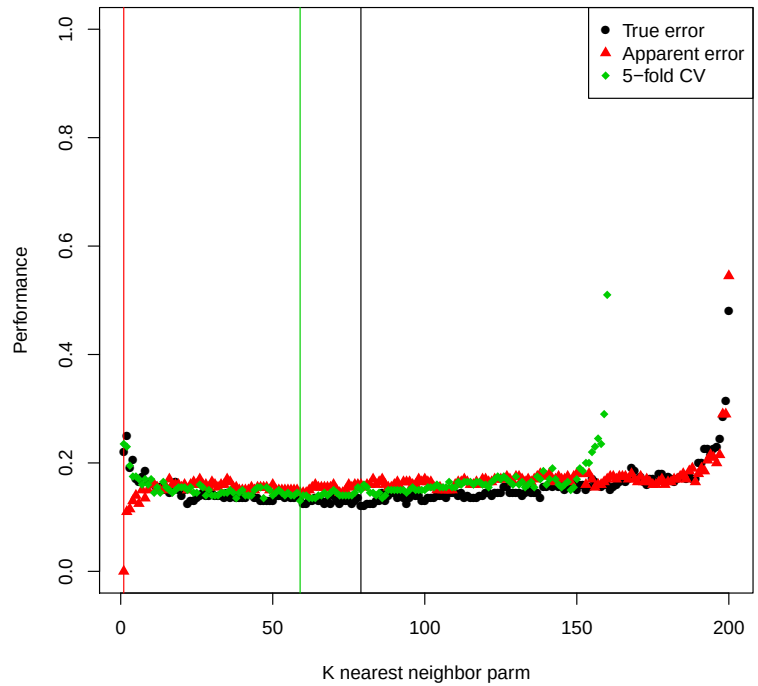
We won't be using the bootstrap for this purpose exactly

However, we will using it extensively when we talk about bagging

Results

Let's return to the earlier examples





Postamble:

- Look at over-fitting

  (If we use the same data to estimate model and tuning parameters, we are likely to overfit)
- Overview data splitting strategies

  (Data splitting works fine with lots of data, but it can be improved upon by resampling)
- Look into resampling techniques

  (We will mainly use CV for judging model performance rather than the bootstrap, but we will use the bootstrap extensively for other purposes)