STAT318 — Data Mining

Dr Thomas Li

University of Canterbury, Christchurch, New Zealand

Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.

Cross-Validation and the Bootstrap

- In this section we discuss two important **resampling methods**: **cross-validation** and the **bootstrap**.

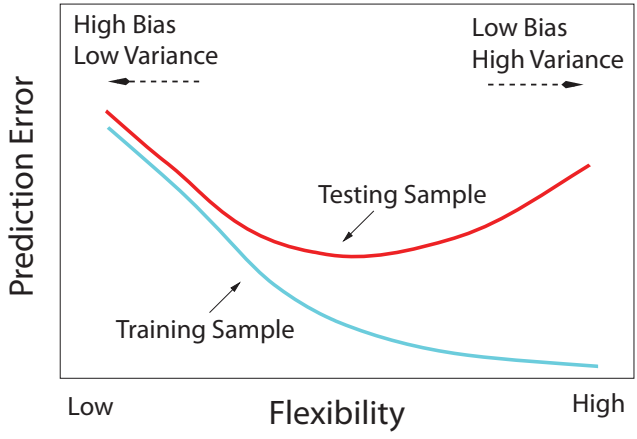
- These methods use samples formed from the training data to obtain additional information about a fitted model or an estimator.

- They can be used for estimating prediction error, determining appropriate model flexibility, estimating standard errors, ...

Training error vs. test error

- The **training error** is the average error that results from applying a statistical learning method to the observations used for training — a simple calculation.
- The **test error** is the average error that results from applying a statistical learning technique to test observations that were not used for training — a simple calculation if test data exists, but we usually only have training data.
- The training error tends to dramatically under-estimate the test error.

Performance



We want to be able to determine the correct level of model flexibility (be near the minimum of the testing error curve) for our particular statistical learning problem.

Validation Set Approach

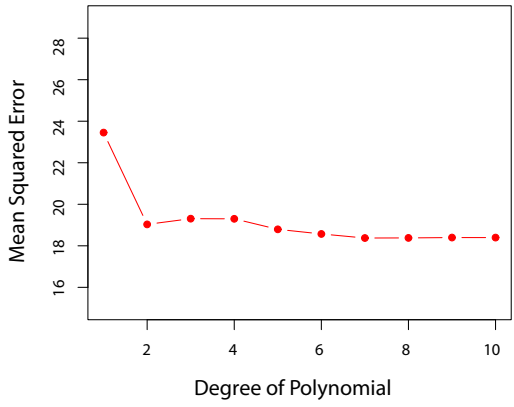
- A very simple strategy is to randomly divide the training data into two sets:
  1. **Training Set:** Fit the model using the training set.
  2. **Validation Set:** Predict the response values for the observations in the validation set.
- The validation set error provides an estimate of the test error (MSE for regression and error rate for classification).

The key word here is **randomly**. You should always randomly split the data to destroy data entry ordering and to ensure that each set captures all the characteristics of the population. You also need to be careful about ephemeral predictors (those whose predictive powers fade away over time). If their predictive powers are strong in the training set and weak in the testing set, poor error estimates can be obtained.

Validation Set Approach

- In this example, the training data are randomly split into two sets of approximately the same size. The blue set is used for training and the orange set for validation.

Example: auto data



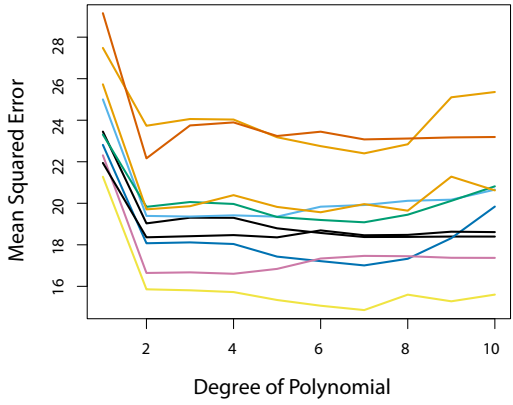
- Find the best level of flexibility in polynomial regression using the validation set approach (50% used for training).

You will recall this example from earlier slides, where we were using horsepower to predict miles per gallon (mpg). We consider polynomial regression models of the form:

$$\text{mpg} = \beta_0 + \sum_{i=1}^{d} \beta_i (\text{horsepower})^i,$$

where d ranges from 1 through to 10. Using the validation set approach, we see that the quadratic model (i=2) is the best model here. Adding higher order polynomial terms to the model does not substantially reduce the test MSE, but it does make the model more complex. We should always choose the most parsimonious model (the simplest model that works well).

Example: auto data



- The validation set approach using different training and validation sets (50% used for training).

We can see that the validation set approach is sensitive to the way the training data are split. The curves have the same shape and suggest the same level of flexibility (quadratic in this case), but the test MSE estimates are highly variable. There are two main reasons for this variability:

- the training data split; and
- only half the observations are used to fit the model.

If the training data set is large, the validation set approach works well because there are enough observations to fit the model and the data split variability tends to be small.

Drawbacks of the validation set approach

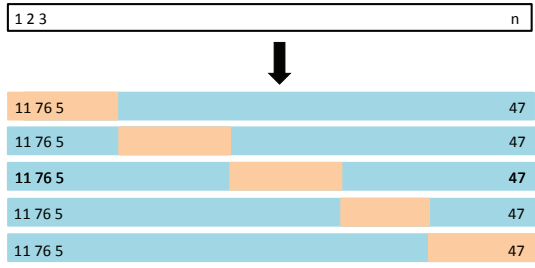
- The validation set estimate of the test error can be highly variable.
- Only a subset of observations are used to train the model. Hence, the validation set test error will tend to over-estimate the test error.

K-fold cross-validation

1. The training data are divided into K groups (folds) of approximately equal size.
2. The first fold is used as a validation set and the remaining $(K-1)$ folds are used for training. The test error is then computed on the validation set.
3. This procedure is repeated K times, using a different fold as the validation set each time.
4. The estimate of the test error is the average of the K test errors from each validation set.

Cross-validation attempts to mimic the validation set approach without the need for a validation set. The training data should be randomly partitioned into K groups to destroy data entry ordering etc.

5-fold cross-validation



- Each row corresponds to one iteration of the algorithm, where the orange set is used for validation and the blue set is used for training.
- If $K = n$, we have **leave one out cross-validation (LOOCV)**.

The LOOCV method can be computationally expensive because you need to fit many models. However, for linear models, the LOOCV test error can be computed from one fit (see page 180 in the course textbook). Best practice (at least according Hastie et al.) is to choose $K = 5$ or 10, possibly averaged over several random data splits.

Cross-validation test error

- We randomly divide the training data of n observations into K groups of approximately equal size, $C_1, C_2, \ldots, C_K$.
- **Regression Problems (average MSE):**

$$\mathrm{CV}_K = \frac{1}{n} \sum_{k=1}^{K} \sum_{i:x_i \in C_k} (y_i - \hat{y}_i)^2.$$

- **Classification Problems (average error rate):**

$$\mathrm{CV}_K = \frac{1}{n} \sum_{k=1}^{K} \sum_{i:x_i \in C_k} I(y_i \neq \hat{y}_i).$$

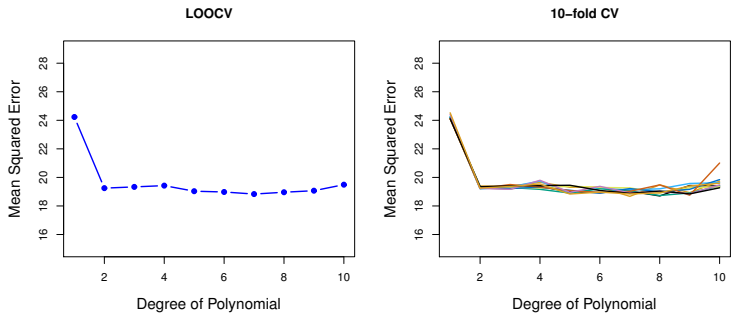
Assuming n/K is an integer (or close enough to an integer), we have

$$\mathrm{CV}_K = \frac{1}{n} \sum_{k=1}^{K} \sum_{i:x_i \in C_k} (y_i - \hat{y}_i)^2 = \frac{1}{K} \sum_{k=1}^{K} MSE_k,$$

where

$$MSE_k = \frac{K}{n} \sum_{i:x_i \in C_k} (y_i - \hat{y}_i)^2.$$

Cross-validation: auto data

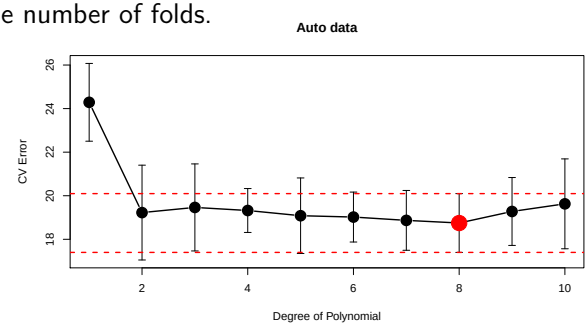


- The right plot shows nine different 10-fold cross-validations.

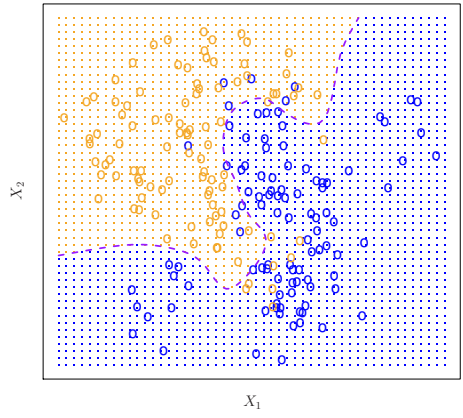
We can see that the variability in the estimated test MSE has been removed (by averaging). The one standard error (SE) rule can be used to choose the best model (although you are not expected to this is this course). We choose the most parsimonious model whose test MSE is within one SE of the minimum test MSE. The SE is computed using

$$\text{SE} = \sqrt{\frac{var(MSE_1, MSE_2, \ldots, MSE_K)}{K}},$$

where K is the number of folds.



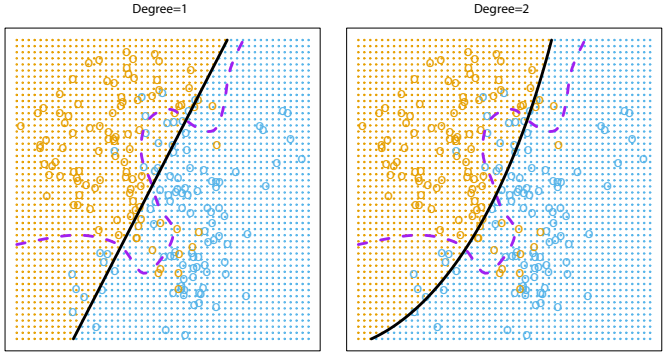
K-fold cross-validation: classification



- The Bayes error rate is 0.133 in this example.

You will recall this example from earlier in the course. where the dashed line is Bayes decision boundary. Bayes error rate is the test error rate of Bayes classifier (the best you can do), which can be calculated here because this is a simulated example.

K-fold cross-validation: logistic regression



- The test error rates are 0.201 and 0.197, respectively.

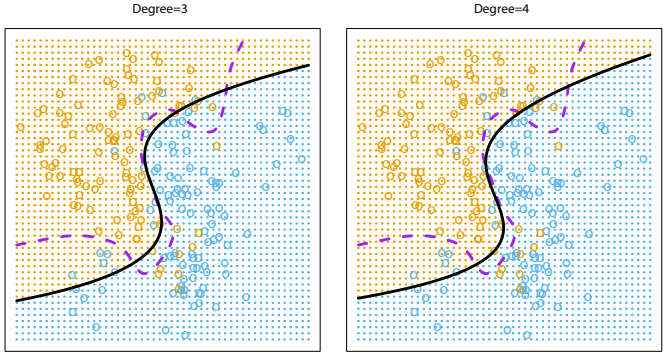
Degree 1:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

Degree 2:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + \hat{\beta}_4 x_2^2$$

K-fold cross-validation: logistic regression



- The test error rates are 0.160 and 0.162, respectively.

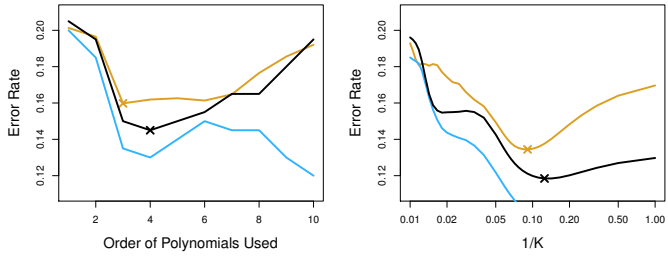
Degree 3:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + \hat{\beta}_4 x_2^2 + \hat{\beta}_5 x_1^3 + \hat{\beta}_6 x_2^3$$

Degree 4:

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1^2 + \hat{\beta}_4 x_2^2 + \hat{\beta}_5 x_1^3 + \hat{\beta}_6 x_2^3 + + \hat{\beta}_7 x_1^4 + \hat{\beta}_8 x_2^4$$

K-fold cross-validation: logistic regression and KNN



- The test error (orange), training error (blue) and the 10-fold cross-validation error (black). The left plot shows logistic regression and the right plot shows KNN.

Recall that Bayes error for this problem was 0.133. As we increase the complexity of the models, we start over-fitting the training error (as expected) and the *characteristic U curve* is visible. The CV error rate does a good job at finding the correct level of flexibility (base of the U) for each model. However, in this example, the CV errors under-estimate the true error rate (but at least we can actually compute the CV error!).

Comments

- Since each training set has approximately $(1 - 1/K)n$ observations, the cross-validation test error will tend to over-estimate the prediction error.
- LOOCV minimizes this upward bias, but this estimate has high variance.
- $K = 5$ or 10 provides a good compromise for this bias-variance trade-off.

LOOCV can have high variance because the training data sets that are used for each fit only differ by one observation (the training folds are highly correlated). Once again, best practice (at least according Hastie et al.) is to choose $K = 5$ or 10, possibly averaged over several random data splits.

Cross Validation: right and wrong

- Consider the following classifier for a two-class problem:
  1. Starting with 1000 predictors and 50 observations, find the 20 predictors having the largest correlation with the response.
  2. Apply a classifier using only these 20 predictors.
- If we use cross-validation to estimate test error, can we simply apply it at step (2)?

The filtering step (subset selection) is a training step because the response variable is used. Hence, we cannot simply apply CV at step (2). We need to apply CV to the full learning process. That is, divide the data into k-folds, find the 20 predictors that are most correlated with the response using $k-1$ of the folds and then fit the classifier to those 20 predictors. We would expect different '20 best predictors' to be found at each iteration and hence, we expect to fit the model using different predictors at each iteration.

Unsupervised screening steps are fine (e.g. choose the 20 predictors with the highest variance across all 50 observations) because the response variable is not used (it's not supervised).

Bootstrap

- The use of the term bootstrap derives from the phrase:

  "to pull oneself up by one's bootstraps".

- The bootstrap is a powerful statistical tool that can be used to quantify uncertainty associated with a statistical learning technique or a given estimator.

Example

- Suppose we wish to invest a fixed sum of money in two financial assets that yield returns X and Y, respectively (X and Y are random variables).

- We want to minimize the risk (variance) of our investment:

$$\min_{\alpha} V(\alpha X + (1-\alpha)Y),$$

where $0 \leq \alpha \leq 1$.

This is a motivating example for the Bootstrap. We can minimise the variance (see below), but we are really just interested in obtaining a statistic. Then we can apply the Bootstrap to investigate properties of our statistic.

Optional material (not assessed) for those interested.

Properties of variance:

$$V(aX) = a^2 V(X)$$
$$V(X+Y) = V(X) + V(Y) + 2\text{cov}(X,Y)$$

Expanding the variance using these properties gives

$$V(\alpha X + (1-\alpha)Y) = \alpha^2 V(X) + (1-\alpha)^2 V(Y) + 2\alpha(1-\alpha)\text{cov}(X,Y).$$

To minimize this expression we set $\frac{d}{d\alpha}V(.) = 0$ which gives

$$2\alpha V(X) - 2(1-\alpha)V(Y) + (2-4\alpha)\text{cov}(X,Y) = 0.$$

Hence, the α that minimises the expression is

$$\alpha = \frac{V(Y) - \text{cov}(X,Y)}{V(X) + V(Y) - 2\text{cov}(X,Y)} = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}.$$

Example

- The values of σ_X^2, σ_Y^2 and σ_{XY} are unknown and hence, need to be estimated from sample data.
- We can then estimate the α value that minimizes the variance of our investment using
$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$
- $\hat{\alpha}$ is an estimator, but we don't know its sampling distribution or its standard error.

Revision material: I hope you know sample variance and sample covariance, but if you don't here they are:

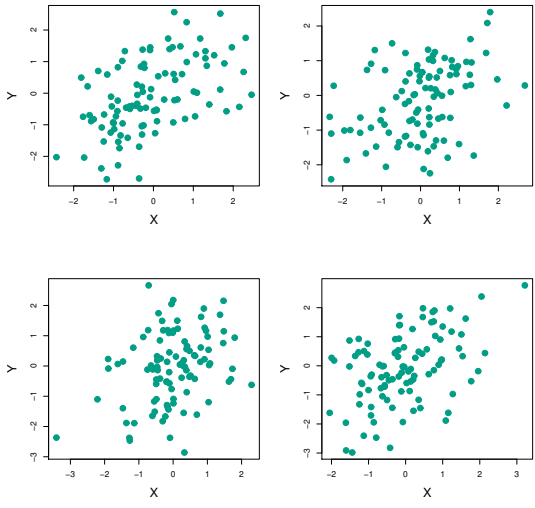
$$\hat{\sigma}_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2;$$

$$\hat{\sigma}_Y^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2;$$

$$\hat{\sigma}_{XY} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}),$$

where $\bar{x}$ and $\bar{y}$ are the sample means for x and y, respectively.

Example: simulated returns for investments X and Y

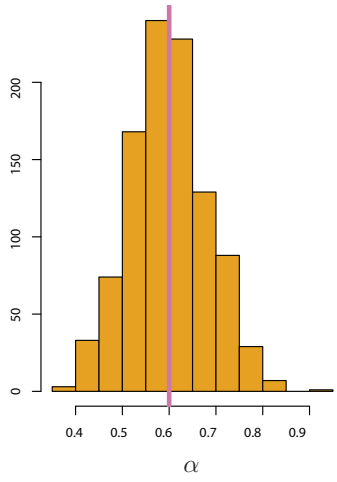


Each plot shows 100 simulated returns for investments X and Y.

- Each sample gives an estimate of α.
- We can use sample statistics to look at statistical properties of our estimator $\hat{\alpha}$, for example, its standard error (standard deviation of an estimator).

The big assumption here is that we can draw as many samples from the population as we like. If we could do this, statistics would be a trivial subject! Clearly, we cannot do this is in practice, we're just trying to illustrate a point here.

Example: simulated sampling distribution for $\hat{\alpha}$



The sampling distribution of $\hat{\alpha}$ using 1000 simulated data sets (a histogram of 1000 $\hat{\alpha}$ values, one $\hat{\alpha}$ from each data set). The true value of α is shown by the vertical line ($\alpha = 0.6$).

Example: statistics from 1000 observations

- The sample mean is

$$\bar{\alpha} = \frac{1}{1000} \sum_{i=1}^{1000} \hat{\alpha}_i = 0.5996,$$

  which is very close to the true value, $\alpha = 0.6$.

- The standard deviation is

$$\text{sd}(\hat{\alpha}) = \sqrt{\frac{1}{1000-1} \sum_{i=1}^{1000} (\hat{\alpha}_i - \bar{\alpha})^2} = 0.083,$$

  which gives an approximate standard error of $\text{SE}(\hat{\alpha}) = 0.083$.

Statistics is very easy if we can draw as many samples from the population as we like!

Real world

- The procedure we have discussed cannot be applied in the *real world* because we cannot sample the original population many times.
- The **bootstrap** comes to the rescue.
- Rather than sampling the population many times directly, we repeatedly sample the observed sample data using **random sampling with replacement**.
- These **bootstrap samples** are the same size as the original sample (n observations) and will likely contain repeated observations.

Bootstrap samples contain approximately $2/3$ of the original training observations (or if you prefer, approximately $1/3$ of the original training observations are not included in a Bootstrap sample). The probability of a training observation x_i being included (at least once) in the bootstrap sample is

$$Pr(x_i \in \text{Bootstrap}) = 1 - (1 - 1/n)^n .$$

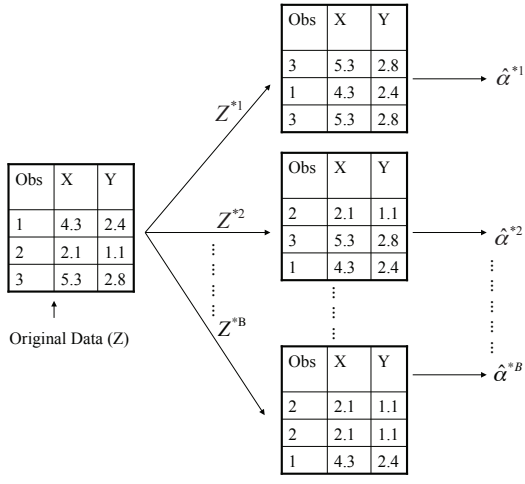
Plugging in some n values we see that

$$Pr(x_i \in \text{Bootstrap}) = 0.6513 \text{ for } n = 10,$$

$$Pr(x_i \in \text{Bootstrap}) = 0.6340 \text{ for } n = 100.$$

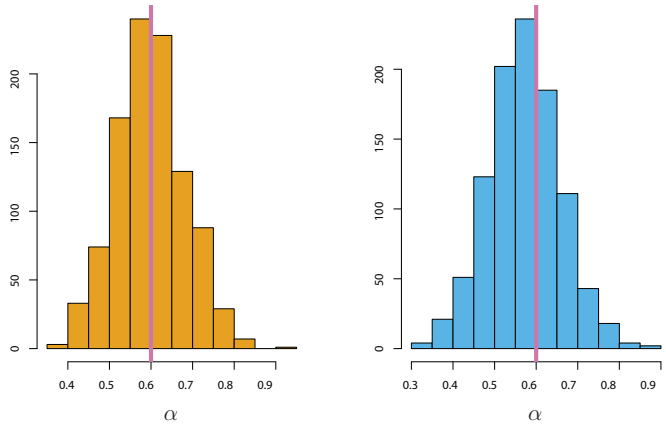
As $n \to \infty$ you can show (not in this course) that

$$\lim_{n \to \infty} 1 - (1 - 1/n)^n = 1 - \exp(-1) \approx 2/3.$$



In this figure, there are only three observations and hence, the Bootstrap is pointless. We are just trying to illustrate the procedure here! That is, each Bootstrap sample gives an estimate of α (the estimator of interest) and if we draw many Bootstrap samples, we get many different estimates of α.

Example: bootstrap sampling distribution for $\hat{\alpha}$



(Left) The simulated sampled distribution for $\hat{\alpha}$. (Right) Histogram showing 1000 bootstrap estimates of α. The bootstrap histogram is similar to the simulated sampling distribution:

- both have similar shape;
- both are centred on $\alpha = 0.6$; and
- both have similar variance.

Hence, we can learn statistical properties of $\hat{\alpha}$, for example its standard error, by looking at the sample statistics from the Bootstrap samples.

Example: statistics from B bootstrap samples

- Let Z^{*i} and $\hat{\alpha}^{*i}$ denote the ith bootstrap sample and the ith bootstrap estimate of α, respectively.
- We estimate the standard error of $\hat{\alpha}$ using

$$\text{SE}_B = \sqrt{\frac{1}{B-1}\sum_{i=1}^{B}(\hat{\alpha}^{*i} - \bar{\alpha}^*)^2},$$

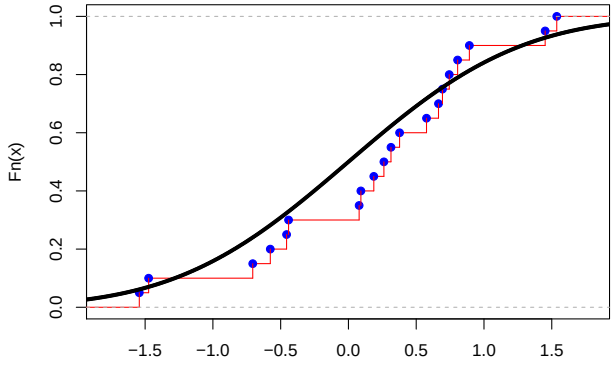
  where B is some large value (say 1000) and

$$\bar{\alpha}^* = \frac{1}{B}\sum_{i=1}^{B}\hat{\alpha}^{*i}.$$

- For our example $\text{SE}_B = 0.087$

The simulated (true) SE was 0.083, so $SE_B \approx SE(\hat{\alpha})$ for this example.

The general picture



- An empirical distribution function (EDF) for a sample of $n = 20$ standard normal random variables.

The EDF is defined as

$$\hat{F}_n(x) = \frac{\#\{x_i < x\}}{n} = \frac{1}{n}\sum_{i=1}^{n} I(x_i < x),$$

where $I(.) = 1$ if the condition is true and zero otherwise (the indicator function).

- The EDF approximates the CDF (there are strong mathematical arguments here that we will not be going into!).
- Sampling from the EDF (the Bootstrap) approximates sampling from the population CDF (the black curve that we want to draw samples from).
- Bootstrap samples will only contain values from the observed sample, but each bootstrap sample will contain different combinations of these values trying to mimic a newly drawn random sample from the population.

Comments

- The bootstrap can also be used to approximate confidence intervals, the simplest method being the **bootstrap percentile** confidence interval.
- For example, an approximate 90% confidence interval is 5th and 95th percentiles of B bootstrap estimates.
- It is possible to use the bootstrap for estimating prediction error, but cross-validation is easier and gives similar results.
- We will use the bootstrap when building decision trees.

For the investment example, the 5th and 95th percentiles of the 1000 $\hat{\alpha}$ values were

$$(0.43, 0.72),$$

which gives an approximate 90% confidence interval for α. To get an approximate 95% confidence interval, for example, we would use the 2.5th and 97.5th percentiles.

There are more advanced Bootstrap methods available (not covered here), but we only need a basic understanding of the Bootstrap. We will only be using the Bootstrap to fit ensemble methods (next section) in this course.