Chapter 5: Resampling Methods

Yu-Tzung Chang and Hsuan-Wei Lee

Department of Political Science, National Taiwan University

2018.11.1.

Outline

- Cross-validation
- Bootstrap

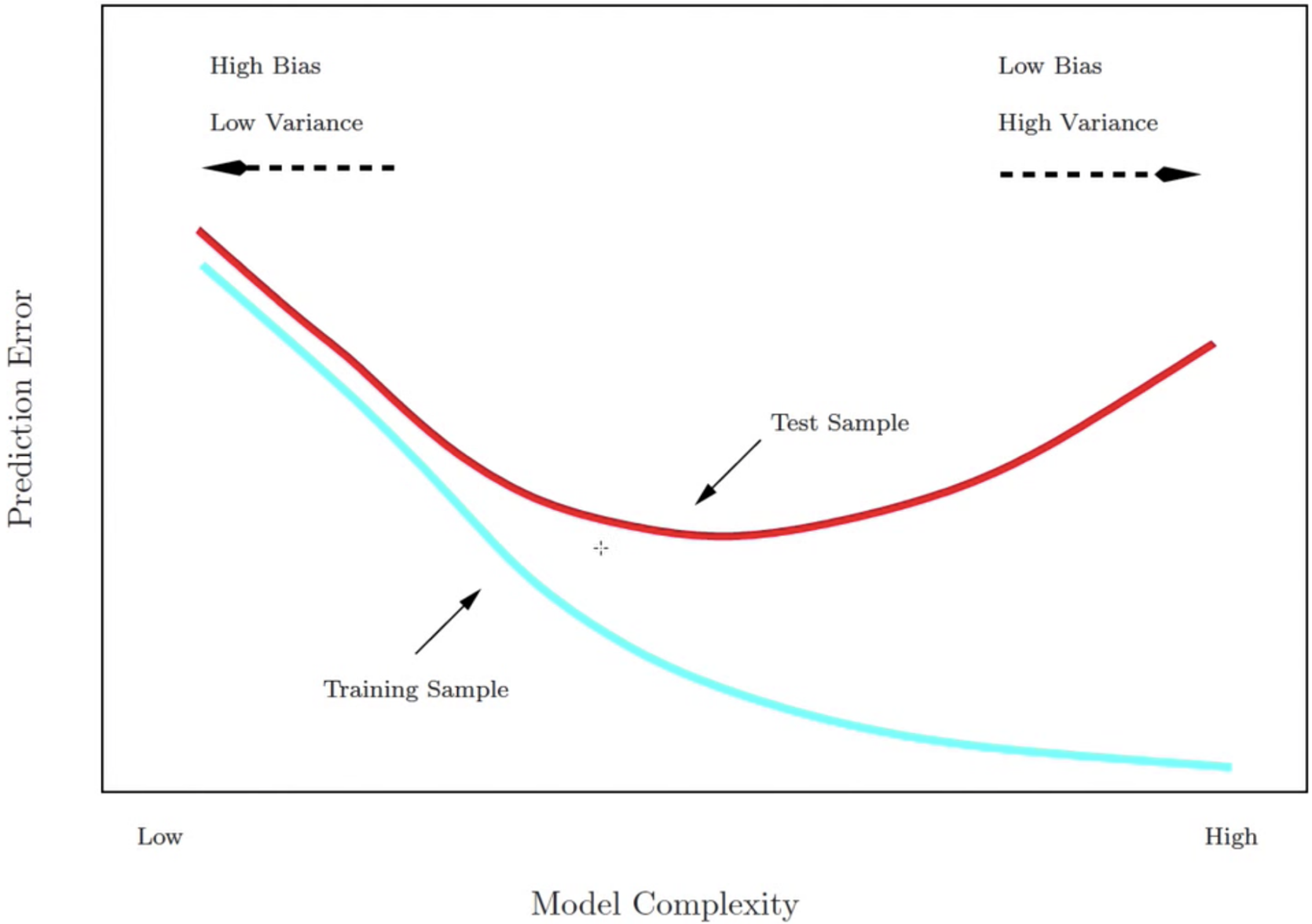
Cross-validation and the bootstrap

- In this section we discuss two *resampling* methods: cross-validation and bootstrap.
- These methods refit a model of interest to samples formed from the training set, in order to obtain additional information about the fitted model.
- For example, they provide estimates of test-set prediction error, and the standard deviation and bias of our parameter estimates.

Training error versus test error

- Recall the distinction between the *test error* and the *training error*.
- The *test error* is the average error that results from using a statistical learning method to predict the response on a new observation, one that was not used in training the method.
- In contrast, the *training error* can be easily calculated by applying the statistical learning method to the observations used in its training.
- But the training error rate often is quite different from the test error rate, and in particular the former can *dramatically underestimate* the error.

Training- versus test-set performance



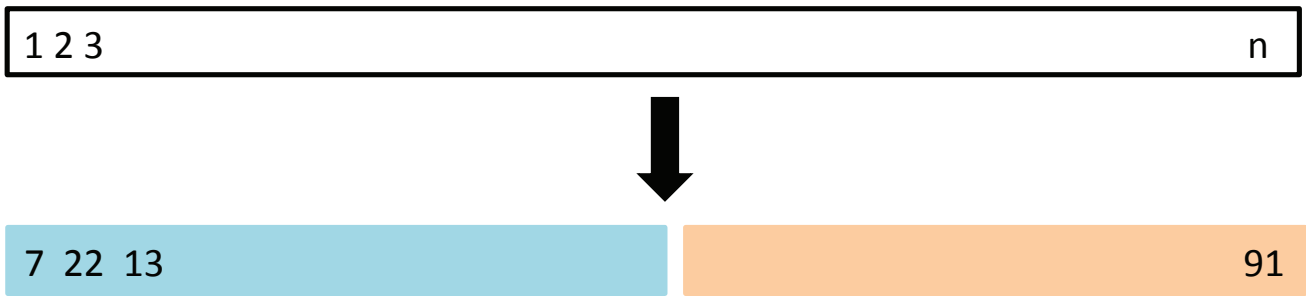
More on prediction-error estimates

- Best solution: a large designed test set. But it's often not available.
- Some methods make a *mathematical adjustment* to the training error rate in order to estimate the test error rate. These include the *Cp statistic*, *AIC* and *BIC*, which we will discuss in the course.
- Here we instead consider a class of methods that estimates the test error by *holding out* a subset of the training observations learning method to those held out observations.

Validation-set approach

- Here we randomly divide the available set of samples into two parts: a *training set* and a *validation* or *hold-out* set.
- The model is fit on the training set, and the fitted model is used to predict the responses for the observations in the validation set.
- The resulting validation-set error provides an estimate of the test error. This is typically assessed using MSE in the case of a quantitative response and misclassification rate in the case of a qualitative (discrete) response.

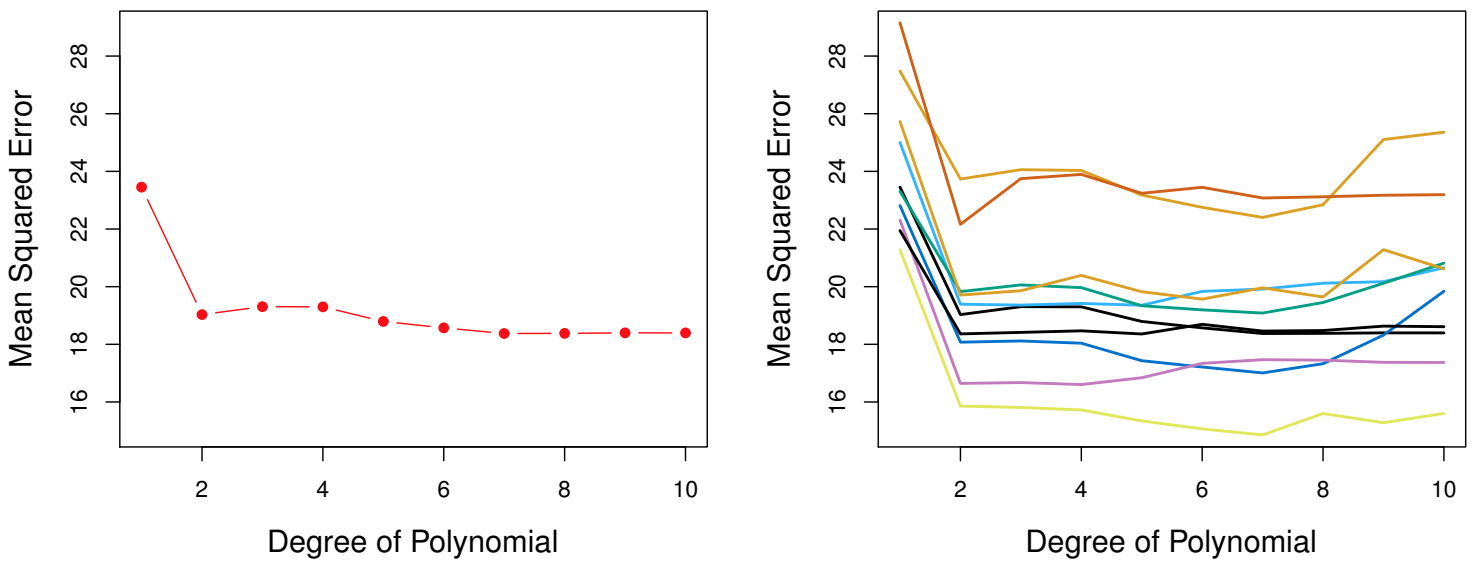
The validation process



A random splitting into two halves: left part is training set, right part is validation set.

Example: automobile data

- Want to compare linear vs higher-order polynomial terms in a linear regression.
- Randomly split the 392 observations into two sets, a training set containing 196 of the data points, and a validation set containing the remaining 196 observations.



Left panel shows single split; right panel shows multiple splits.

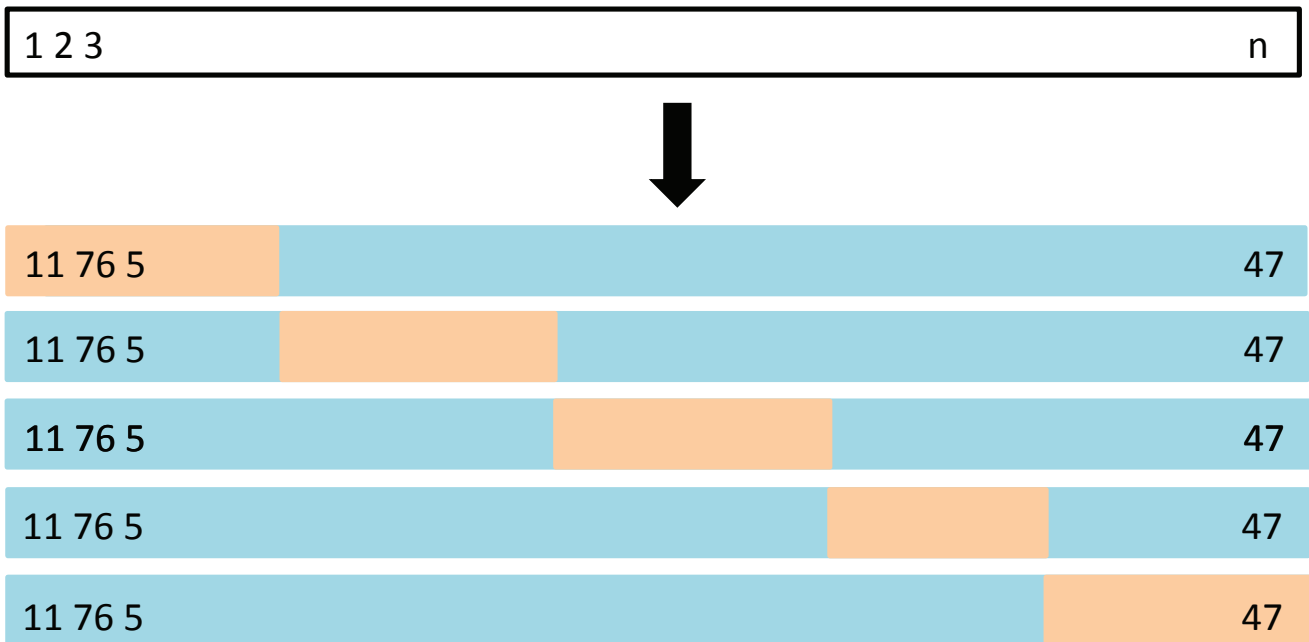
Drawbacks of validation set approach

- The validation estimate of the test error can be highly variable, depending on precisely which observations are included in the training set and which observations are included in the validation set.
- In the validation approach, only a subset of the observations – those that are included in the training set rather than in the validations set – are used to fit the model.
- This suggests that the validation set error my tend to *overestimate* the test error for the model fit on the entire data set.

K-fold cross-validation

- Widely used approach for estimating test error.
- Estimates can be used to select best model, and to give an idea of the test error of the final chosen model.
- Idea is to randomly divide the data into K equal-sized parts. We leave out part k, fit the model to the other $K-1$ parts (combined), and then obtain predictions for the left-out kth part.
- This is done in turn for each part $k = 1, 2, \ldots, K$, and then the results are combined.

Example: 5-fold cross-validation



The details

- Let the K parts be $C_1, C_2, \ldots, C_K$, where C_k denotes the indices of the observations in part k. There are n_k observations in part k: if N is a multiple of K, them $n_k = n/K$.
- Compute

$$CV_K = \sum_{k=1}^{K} \frac{n_k}{n} MSE_k$$

  where $MSE_k = \sum_{i \in C_k} (y_i - \hat{y}_i)^2 / n_k$, and $\hat{y}_i$ is the fit for observation i, obtained from the data with part k removed.
- Setting $K = n$ yields n-fole or *leave-one-out-cross-validation* (LOOCV).

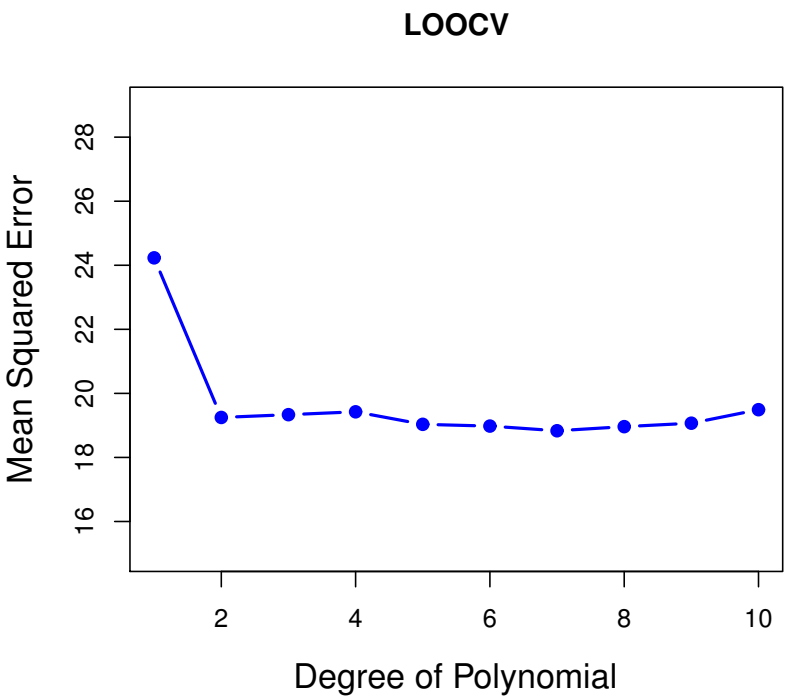
A nice special case

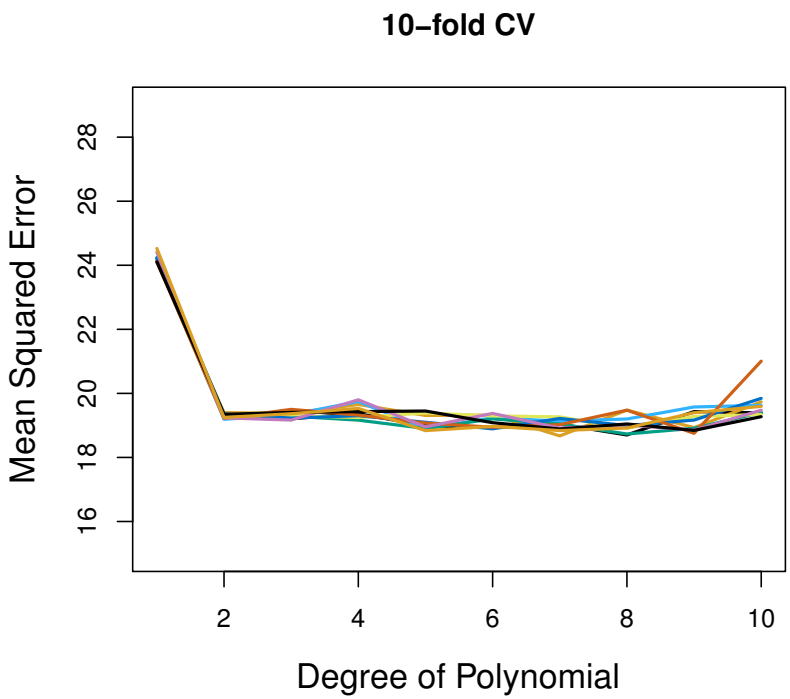
- With least-squares linear or polynomial regression, an amazing shortcut makes the cost of LOOCV the same as that of a single model fit! The following formula holds:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2,$$

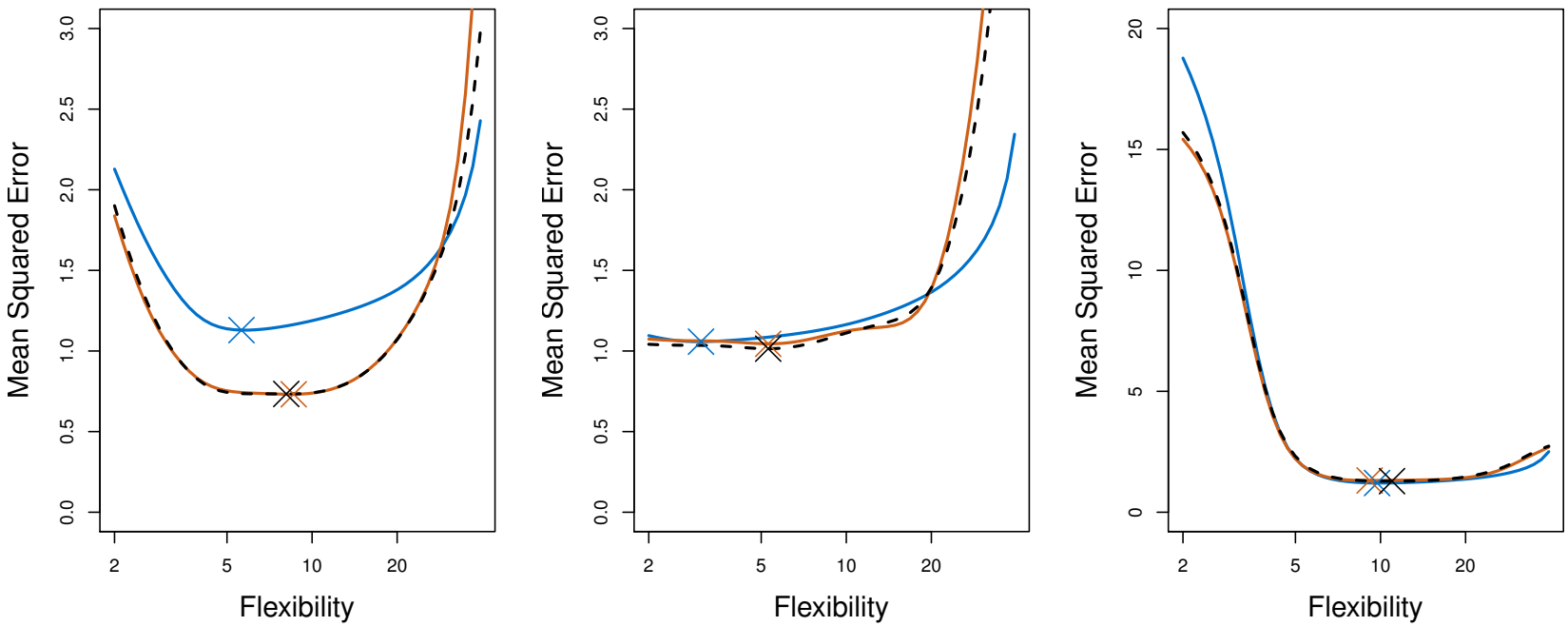
  where $\hat{y}_i$ is the ith fitted value from the original lease squares fit, and h_i is the leverage. This is like the ordinary MSE, except the ith residual is divided by $1 - h_i$.
- LOOCV is sometimes useful, but typically doesn't shake up the data enough. The estimates from each fold are highly correlated and hence their average can have high variance.
- A better choice is $K = 5$ or 10.

Auto data revisited





True and estimated test MSE for the simulated data



Other issues with cross-validation

- Since each training set is only $(K-1)/K$ as big as the original training set, the estimates of prediction error will typically be biased upward.
- This bias is minimized when $K = n$ (LOOCV), but this estimate has high variance.
- $K = 5$ or 10 provides a good compromise for this bias-variance tradeoff.

Cross-validation for classification problems

- We divide the data into K roughly equal-sized parts $C_1, C_2, \ldots, C_K$ denotes the indices of the observations in part k. There are n_k observations in part k: in n is a multiple of K, then $n_k = n/K$.
- Compute

$$CV_K = \sum_{k=1}^{K} \frac{n_k}{n} Err_k$$

  where $Err_k = \sum_{i \in C_k} I(y_i \neq \hat{y}_i)/n_k$.
- The estimated standard deviation of CV_K is

$$\hat{SE}(CV_K) = \sqrt{\sum_{k=1}^{K} (Err_k - \bar{Err}_k)^2/(K-1)}.$$

- This is a useful estimate, but strictly speaking, not quite accurate.

Cross-validation: right and wrong

- Consider a simple classifier applied to some two-class data:
  1. Starting with 5000 predictors and 50 samples, find the 100 predictors having the largest correlation with the class labels.
  2. We then apply a classifier such as logistic regression, using only these 100 predictors.
- How do we estimate the test set performance of this classifier?
- Can we apply cross-validation in step 2, forgetting about step 1?

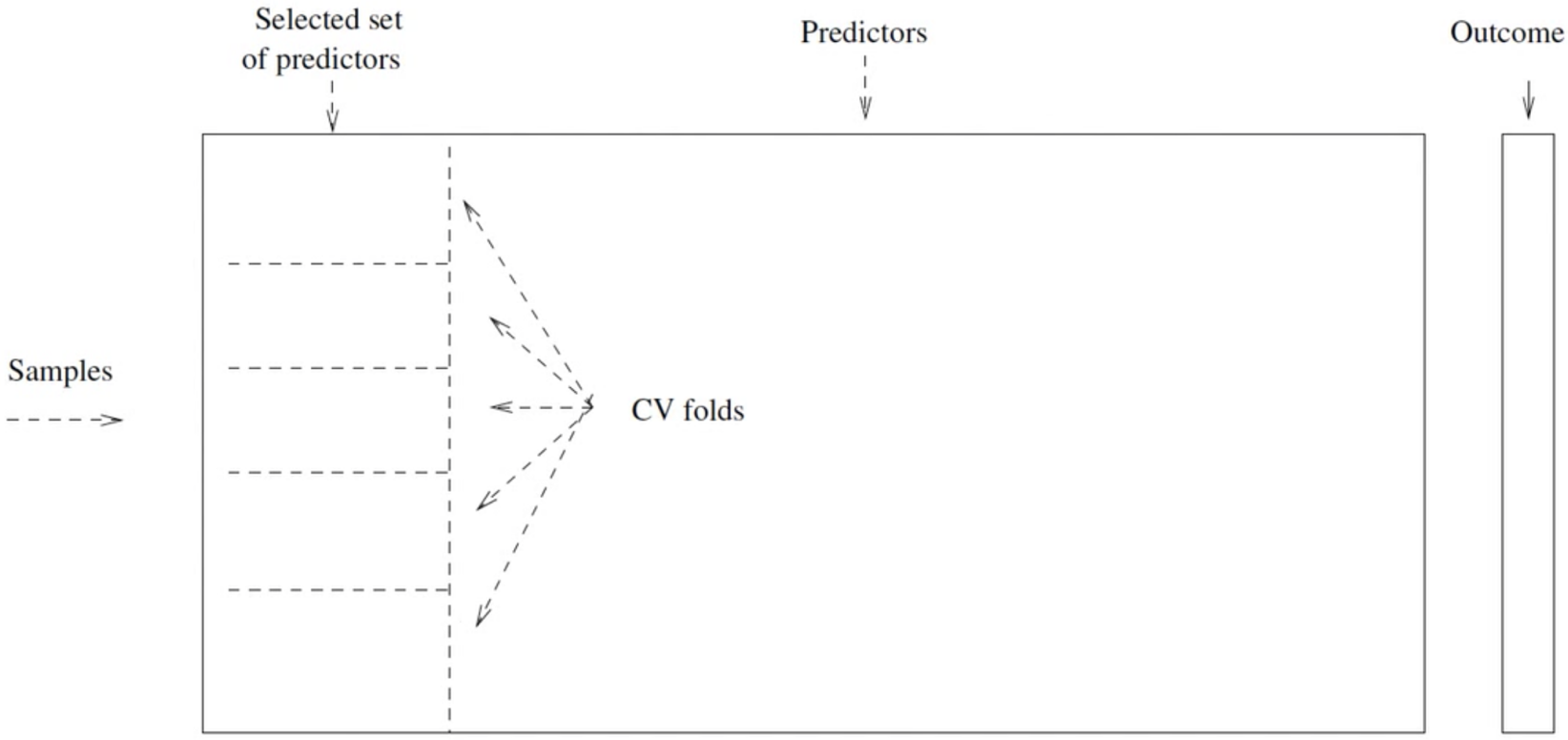
No!

- This would ignore the fact that in Step 1, the procedure *has already seen the labels of the training data*, and made use of them. This is a form of training and must be included in the validation process.
- It is easy to simulate realistic data with the class labels independent of the outcome, so that true test error $= 50\%$, but the CV error estimate that ignores Step 1 is zero!
- This issue exists in many high profile genomics papers.

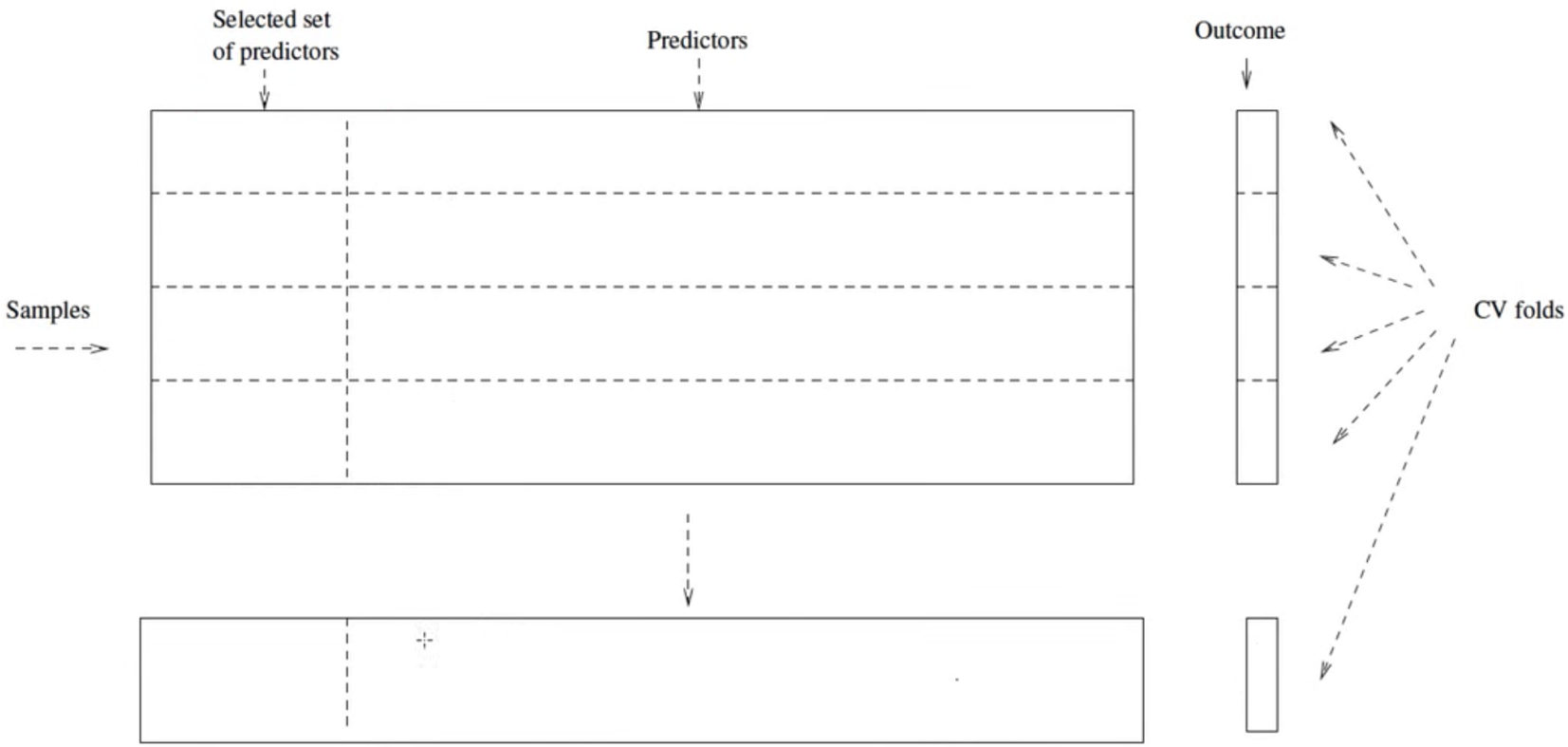
The wrong and the right way

- Wrong: Apply cross-validation in step 2.
- Right: Apply cross-validation in step 1 and 2.

Wrong way


Selected set
of predictors
Predictors
Outcome
Samples
CV folds

Right way


Selected set
of predictors
Predictors
Outcome
Samples
CV folds

The bootstrap

- The *bootstrap* is a flexible and powerful statistical tool that can be used to quantify the uncertainty associated with a given estimator or statistical learning method.
- For example, it can provide an estimate of the standard error of a coefficient, or a confidence interval for that coefficient.

Where does the name come from?

- The use of the term bootstrap derives from the phrase *to pull oneself up by one's bootstrap*, widely though to be based on one of the eighteenth century "The Surprising Adventures of Baron Munchausen" by Rudolph Erich Raspe:
  The Baron had fallen to the bottom of a deep lake. Just when it looked like all was lost, ht though to pick himself up by his own bootstraps.
- It is not the same as the term "bootstrap" used in computer science meaning to "boot" a computer from a set of core instructions, though the derivation is similar.

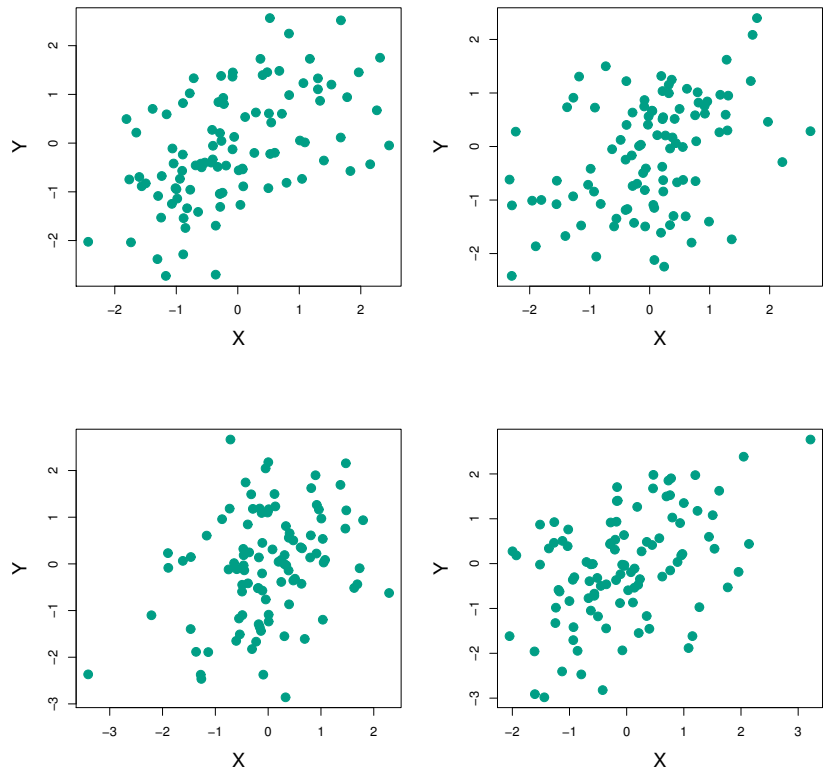
A simple example

- Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, respectively, where X and Y are random quantities.
- We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.
- We wish to choose α to minimize the total risk, or variance, of our investment. In other words, we want to minimize $Var(\alpha X + (1 - \alpha Y))$.
- One can show that the value that minimizes the risk is given by

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

where $\sigma_X^2 = Var(X)$, $\sigma_Y^2 = Var(Y)$, and $\sigma_{XY} = Cov(X, Y)$.

A simple example (continued)



Each panel displays 100 simulated returns of investments X and Y. From left to right and top to bottom, the resulting estimates for α are 0.576, 0.532, 0.657 and 0.651.

A simple example (continued)

- To estimate the standard deviation of $\hat{\alpha}$, we repeated the process of simulating 100 paired observations of X and Y, and estimating α 1,000 times.
- We thereby obtained 1,000 estimates for α, which we can call $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$.
- For these simulations the parameters were set to $\sigma_X^2 = 1$, $\sigma_Y^2 = 1.25$, and $\sigma_{XY} = 0.5$, and so we know that the true value of α is 0.6.

A simple example (continued)

- The mean over all 1,000 estimates for α is

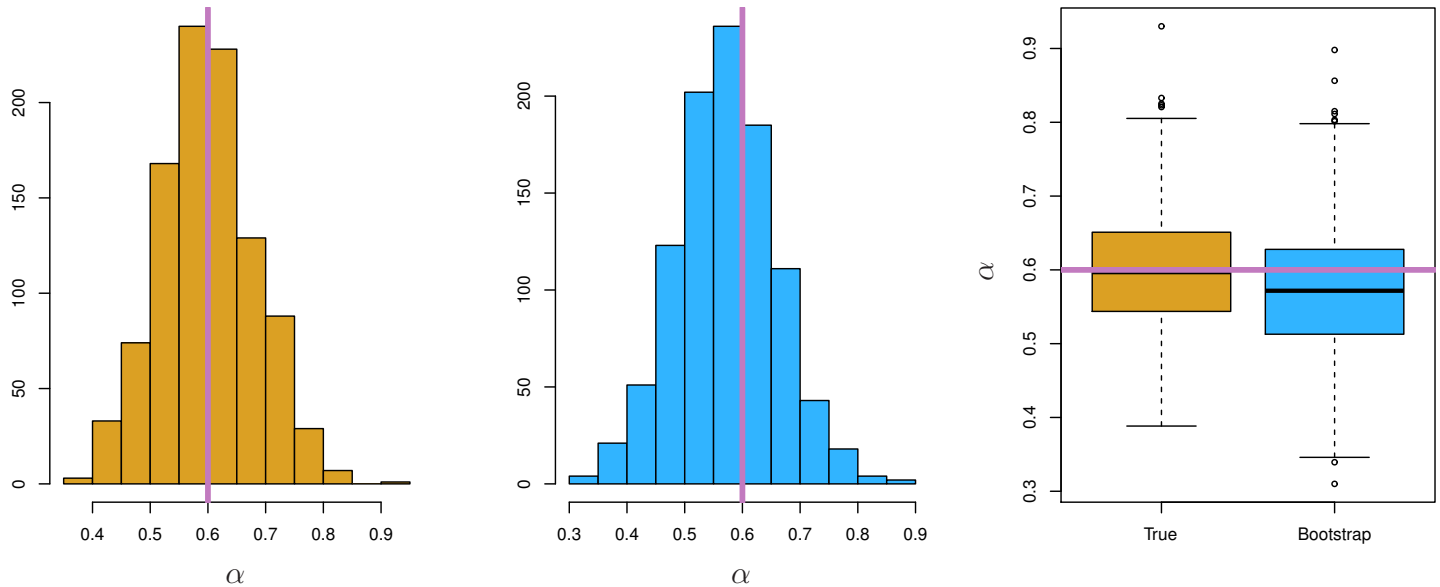
$$\bar{\alpha} = \frac{1}{1000} \sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996,$$

very close to the true $\alpha = 0.6$, and the standard deviation estimate is

$$\sqrt{\frac{1}{1000-1} \sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083.$$

- This gives us a very good idea of the accuracy of $\hat{\alpha}$: $SE(\hat{\alpha}) \approx 0.083$.
- Roughly speaking, for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08, on average.

A simple example (continued)

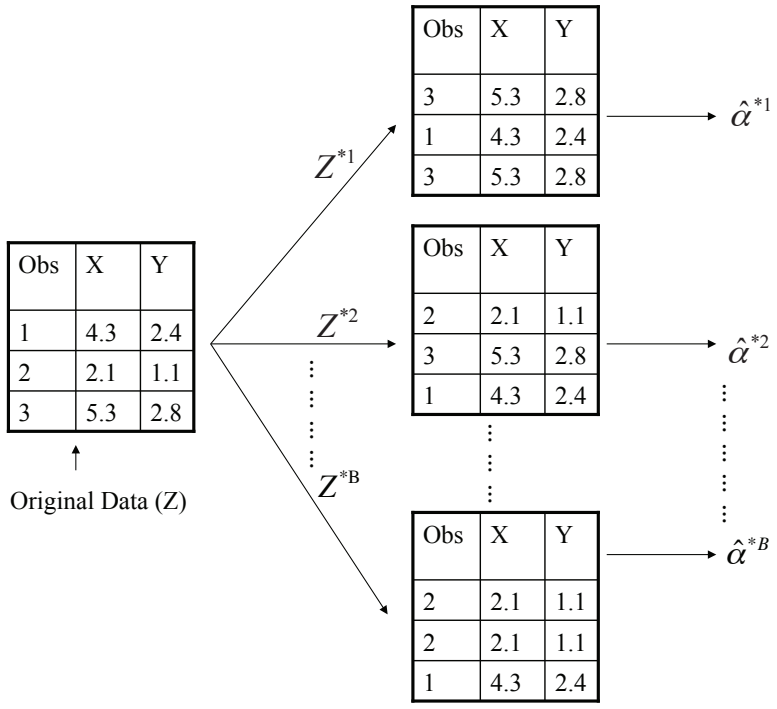


- Left: A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.
- Center: A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.
- Right: The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.

Now back to the real world

- The procedure outlined above cannot be applied, because for real data we cannot generate new samples from the original population.
- However, 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 repeated 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 dataset. As a result some observations may appear more than once in a given bootstrap data set and some not at all.

Example with just 3 observations



A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtained an estimate of α.

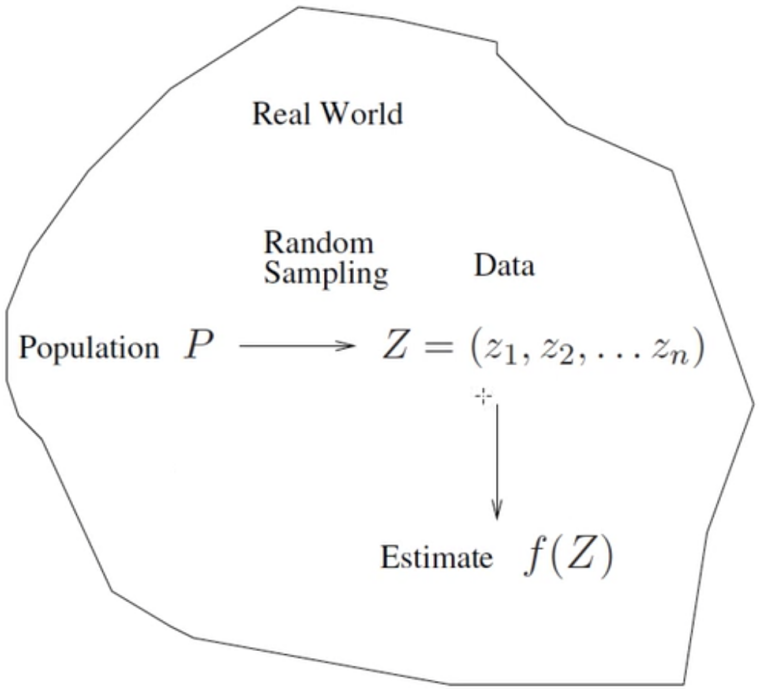
A simple example (continued)

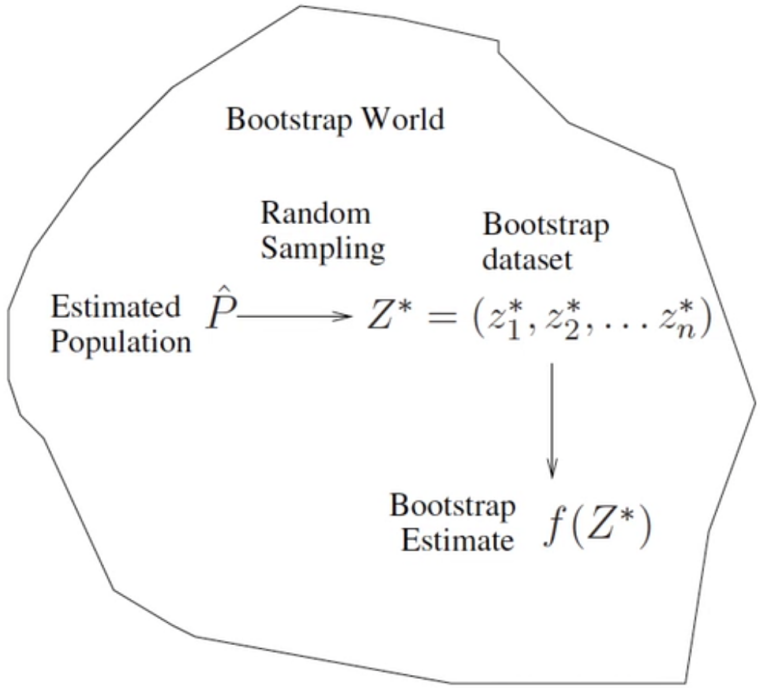
- Denoting the first bootstrap data set by Z^{*1}, we use Z^{*1} to produce a new bootstrap estimate for α, which we call $\hat{\alpha}^{*1}$.
- This procedure is repeated B times for some large number B (say 100 or 1,000), in order to produce B different bootstrap data sets, $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$, and B corresponding α estimates, $\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}$.
- We estimate the standard error of these bootstrap estimates using the formula

$$SE_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{*r} - \bar{\hat{\alpha}}^{*})^2}.$$

- This serves as an estimate of the standard error of $\hat{\alpha}$ estimated from the original data set. For this example $SE_B(\hat{\alpha}) = 0.087$ (compared with the true $SE(\hat{\alpha}) \approx 0.083$).

A general picture for the bootstrap





The bootstrap in general

- In more complex situations, figuring out the appropriate way to generate bootstrap samples can require some thought.
- For example, if the data is a time series, we can't simply sample the observations with replacement.
- We can instead create blocks of consecutive observations, and sample those with replacements. Then we paste together sampled blocks to obtain a bootstrap dataset.

Other uses of the bootstrap

- Primarily used to obtain standard errors of an estimate.
- Also provides approximate confidence intervals for a population parameter. For example, looking at the histogram in the middle panel of the previous results slide, the 5% and 95% quantiles for the 1,000 values is $(0.43, 0.72)$.
- This represents an approximate 90% confidence interval for the true α.
- The above interval is called a *bootstrap percentile* confidence interval. It is the simplest method (among many approaches) for obtaining a confidence interval from the bootstrap.

Can we use bootstrap estimate prediction error?

- In cross-validation, each of the K validation folds is distinct from the other $K-1$ folds used for training: *there is no overlap*. This is crucial for its success.
- To estimate prediction error using the bootstrap, we could think about using each bootstrap dataset as our training sample, and the original sample as our validation sample. But then each bootstrap sample has significant overlap with the original data. About two-thirds of the original data points appear in each bootstrap sample.