STA 5820
Chapter 5
Resampling Methods

Kazuhiko Shinki

Wayne State University

Overview:

The resampling method is a statistical method to sample from the existing data set. It is useful to evaluate the variability of data, and the variability of estimated statistical models.

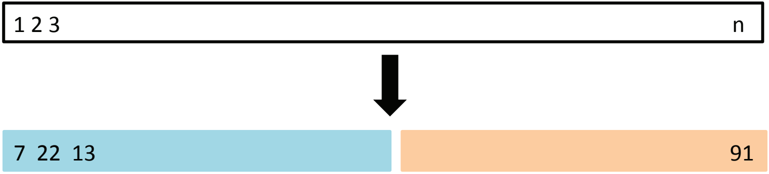
- Cross-Validation: A method to evaluate test errors with multiple randomized training and test sets (5.1).
- Bootstrap : A method to estimate the distribution of parameter estimates with multiple sampling from the data set (5.2).

5.1.1 Validation Set Approach

A review: validation set approach

The simplest method to evaluate the prediction performance is the validation set approach.

1. The data set is divided into the training set and the test (or validation) set.
2. Estimate a statistical model with the training set.
3. Evaluate the prediction error with the test set.



5.1.1 Validation Set Approach

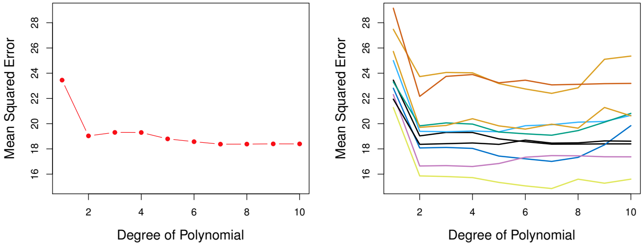
There is a trade-off for the accuracy of statistical model and the accuracy of test error.

The estimated model is more accurate when the training set is large, but the test error is less accurate since the remaining test error is small.

A small training set and a large test set makes the opposite problem.

5.1.1 Validation Set Approach

The data set is divided into the training set and the test set randomly. This randomization produces variation in the test error.

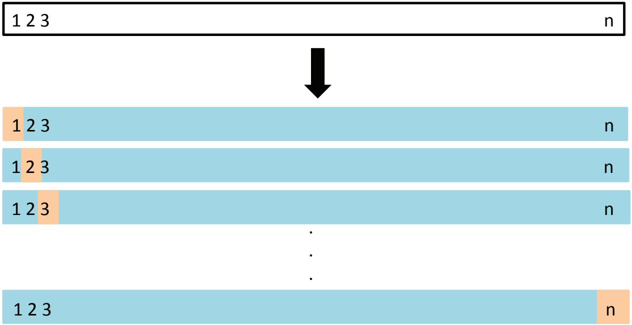


The test error when we regress `mpg` on a polynomial of `horsepower` in the `Auto` data. (Left:) One pair of training and test sets. (Right:) 10 pairs of training and test sets.

5.1.2 Leave-one-out Cross Validation

Idea

Leave-one-out cross-validation (LOOCV) is a repeated procedure to estimate test error.



Blue: training sets; orange: test sets.

5.1.2 Leave-one-out Cross Validation

Algorithm

Suppose $(x_1, y_1), \cdots, (x_n, y_n)$ are observations. Then, LOOCV

1. splits the data set into a training set with $(n-1)$ observations and a test set with the other observation. Suppose that $\{(x_i, y_i)\}$ is the test set.
2. estimates the model with the training set. Let $\hat{y}_i$ is the predicted y_i. The the test error is $(y_i - \hat{y}_i)^2$. (Denote this as MSE_i, the i-th mean square error).
3. repeats the steps 1-2 to calculate MSE_i for all possible i's.
4. calculates the cross-validated error ($CV_{(n)}$) as the mean of MSE_i ($1 \leq i \leq n$).

5.1.2 Leave-one-out Cross Validation

Pros:

- The training set is large, so the estimated model is as accurate as the estimated model with all observations. That is, the bias of the CV error (against the true test error) is very small.
- The training set is large, so the model has less identification problems.
  - For example, suppose a binary predictor value is 0 for most observations and 1 for only a few observations. If the training set is small, the predictor value may be 0 for ALL training observations, resulting an identification problem for the coefficient of the predictor.
- There is no randomization in the method.

Cons:

- Computational burden is large.
  - If there are 1,000 observations, we have to estimate the model 1,000 times.

5.1.2 Leave-one-out Cross Validation

For linear regression models, we have a short cut to calculate $CV_{(n)}$, the LOOCV error.

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

where h_i is

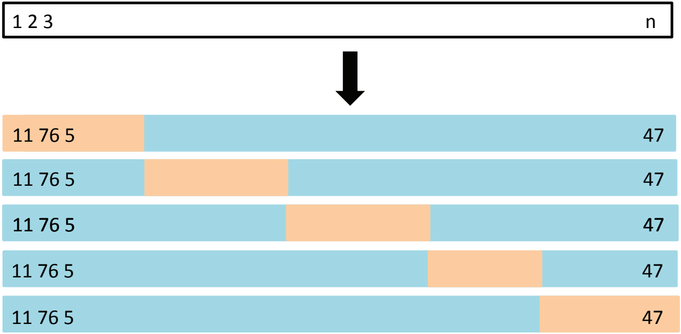
$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^{n} (x_j - \bar{x})^2}.$$

The computational burden is just the same order as fitting a linear regression.

5.1.3 *k*-fold Cross-Validation

Another method of cross-validation is ***k*-fold CV**.

Idea



Blue: training sets; orange: test sets.

5.1.3 k-fold Cross-Validation

Algorithm

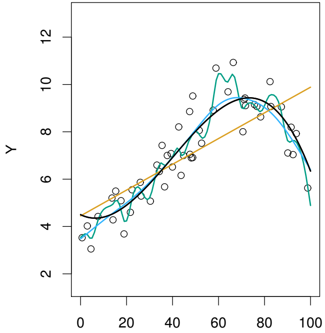
The k-fold CV

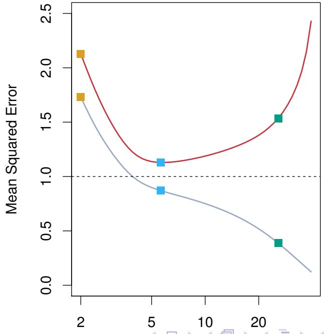
1. divides the data set into k equal sized groups randomly,
2. use the i-th groups as the test set and the remaining $(k-1)$ groups as the training set, and evaluate the MSE (say, MSE_i).
3. repeat the step 2 for all i's.
4. calculate the cross-validated error ($CV_{(k)}$) as the mean of MSE_i ($1 \leq i \leq k$):

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i.$$

5.1.3 *k*-fold Cross-Validation

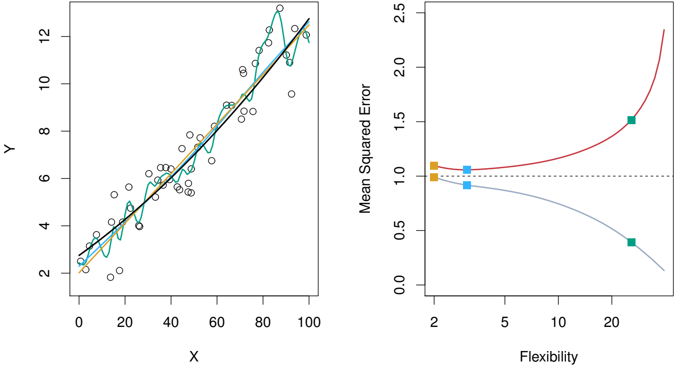
Example 1: The left figure shows simulated data (circles) from the true curve (black) and estimated polynomial curves (yellow, blue and green). The right figure shows the training error (gray) and and the test error (red). Flexibility is the degrees of freedom for the polynomial (2 means a polynomial with degree 1).





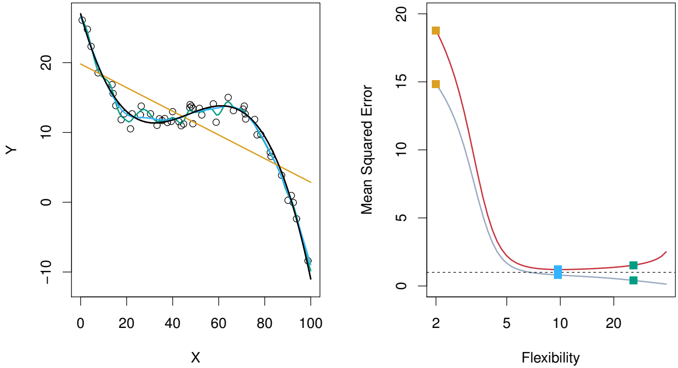
5.1.3 *k*-fold Cross-Validation

Example 2:



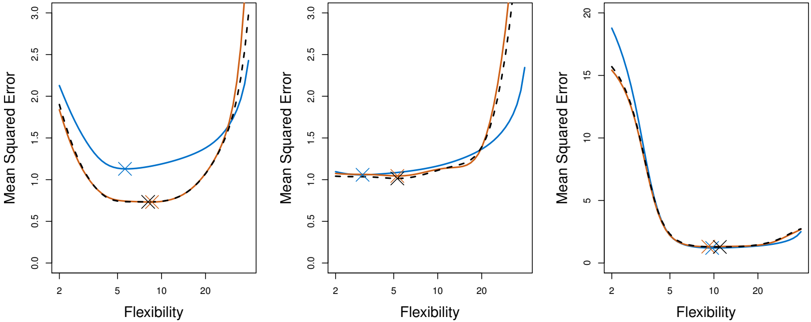
5.1.3 *k*-fold Cross-Validation

Example 3:



5.1.4 Bias-Variance trade-off for *k*-fold Cross-Validation

Results: Examples 1-3 from left to right. The blue curve is true error rate, the black dashed curve is the LOOCV error, and the orange curve is 10-fold CV error.



5.1.4 Bias-Variance trade-off for k-fold Cross-Validation

The LOOCV is a special case of k-fold CV with $k = n$.

How to choose k?

- People often use $k = 5$ or 10. The R default is often $k = 10$.
- A larger k gives a smaller bias to estimate the test error.
- A larger k takes more computational time.
- A larger k has a larger variance to estiamte the test error, because the training sets have excessive similarities for a large k.

5.1.5 Cross-Validation on Classification Problems

CV for classification can be done in the same way as for regression, except that we use error rate and not MSE. For an example of LOOCV,

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} Err_i$$

where $Err_i = I(y_i \neq \hat{y}_i)$, that is, the error is 0 if the classification is correct; the error is 1 if the classification is incorrect.

5.1.5 Cross-Validation on Classification Problems

Example:

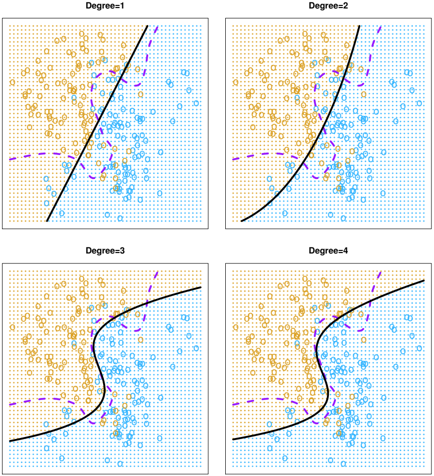
In a logistic regression model

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \cdots + \beta_k X_1^k + \alpha_1 X_2 + \cdots + \alpha_k X_2^k,$$

we want to determine the optimal ***k*** for prediction.

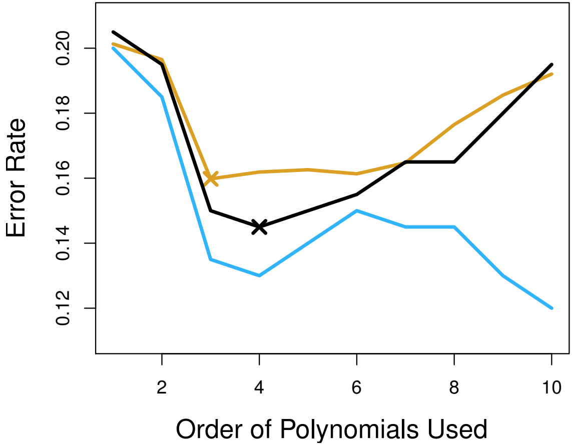
5.1.5 Cross-Validation on Classification Problems

Graphs for $\boldsymbol{k} = \mathbf{1, 2, 3, 4}$: True Bayes decision boundaries (purple dashed) and estimated boundaries by logistic decision (black).



5.1.5 Cross-Validation on Classification Problems

Figure 5-8L: Training error (blue), test error (brown), and 10-fold CV error (black).



5.2 Bootstrap

The bootstrap is a method to estimate the distribution of parameter estimates with multiple sampling from the data set.

5.2 Bootstrap

Suppose that there are $\boldsymbol{n}$ independent observations in the sample, and we want to estimate a parameter α (for example, population mean, population variance, slope of regression line etc.).

The procedure of bootstrap is as follows.

1. Randomly sample $\boldsymbol{n}$ observations with replacement.
2. Calculate the parameter estimate (say, $\hat{\alpha}_{\boldsymbol{i}}$) from the sample in the step 1.
3. Repeat the steps 1-2 for many times (say, $\boldsymbol{B}$ times).
4. Make a distribution of the $\alpha_{\mathbf{1}}, \cdots, \alpha_{\boldsymbol{B}}$.

5.2 Bootstrap

If you want to calculate $\mathbf{Var}(\hat{\alpha})$, the estimate is obtained by bootstrap as

$$\frac{1}{B-1}\sum_{i=1}^{B}(\alpha_i - \bar{\alpha})^2.$$

5.2 Bootstrap

Sampling with replacement means that we sample each observation from the original set so that we may choose the same observations more than once.

R code

```
> sample(c(1,2,4,8,16), replace=T)
[1] 2 4 1 4 8
> sample(c(1,2,4,8,16), replace=T)
[1] 16  8  4  4  4
```

5.2 Bootstrap

Example:

We have data with 5 observations: 5.1, 4.8, 3.9, 5.3, 4.1.

1. Calculate mean and S.D. of the data.
2. Estimate the standard error of the mean theoretically.
3. Estimate the standard error of hte mean by bootstrap.

5.2 Bootstrap

Sample Code:

```
# (1)
> X <- c(5.1, 4.8, 3.9, 5.3, 4.1)
> c(mean(X), sd(X))
[1] 4.640000 0.614817

# (2)
> sd(X)/sqrt(5)   # (standard deviation)/(sqrt of sample size) by CLT
[1] 0.2749545

# (3)
> M <- numeric(100)  # 100-dim vector
> for (i in 1:100){
+ M[i] <- mean(sample(X, replace=T))  # We simulate sample mean 100 times
+ }
> sd(M) # standard deviation of the mean
[1] 0.2549166
```

5.2 Bootstrap

Note on (2):

Recall that, when $X_1, \cdots, X_n$ are given, the standard error (or standard deviation) of the mean $\bar{X}$ is given by:

$$SE_{\bar{X}} = \frac{S.D. \text{ of } X}{\sqrt{n}} \tag{1}$$

It implies that when n is large, the variability of the mean $\bar{X}$ becomes arbitrarily small. This coincides with accepted fact that the mean of many observations is stable.

5.2 Bootstrap

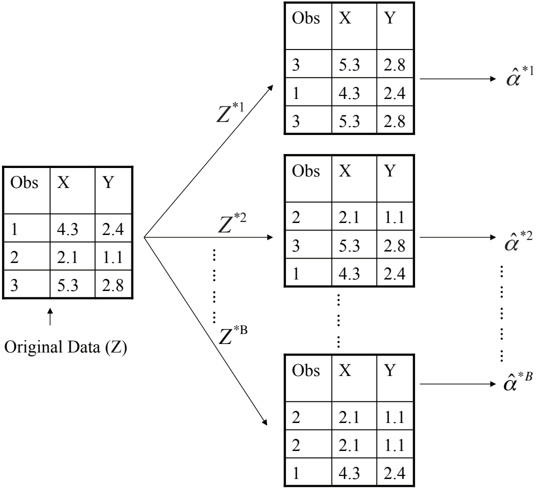
Note on (3):

The estimate (0.2549166) tends to be slightly smaller than the answer in (2), but still a good estimate. The bias becomes negligible when the sample size becomes large.

Bootstrapping is particularly useful when the theoretical distribution of the estimate is very hard to obtain.
Unlike the theoretical distribution of an estimate, bootstrapping can be applied in any estimation problems as long as observations are independent.

5.2 Bootstrap

A general case: Bootstrap can be applied for any estimate α in a statistical model.



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Original Data (Z)

Z^{*1}

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

$\hat{\alpha}^{*1}$

Z^{*2}

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

$\hat{\alpha}^{*2}$

Z^{*B}

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

$\hat{\alpha}^{*B}$

Example

Suppose that there are 200 observations for the response time to a signal ($\boldsymbol{Y}$) and alcohol content in blood ($\boldsymbol{X}$), but some observations come from the same person so there are only 82 persons in the data set. We want to estimate the regression line $\boldsymbol{Y} = \alpha + \beta \boldsymbol{X}$. α and β are estimated by least-square, but the formula for $\mathbf{Var}(\hat{\alpha})$ and $\mathbf{Var}(\hat{\beta})$ do not work becaus observations are dependent.

How to estimate the variability of α and β?

Bootstrap can be used. Randomly sample 82 persons with replacement each time by bootstrapping.
$\mathbf{Var}(\hat{\beta})$ is estimated by bootstrapped estimates: $\hat{\beta}_1, \cdots, \hat{\beta}_B$.

Memo