Lecture 6
Resampling Methods

CHUNG-MING KUAN

Department of Finance & CRETA

National Taiwan University

March 27, 2022

Lecture Outline

1. Introduction

2. Cross Validation
    - The Validation Set Approach
    - Leave-One-Out Cross Validation
    - k-Fold Cross-Validation
    - Comparison between Different Methods

3. Bootstrap
    - Bootstrapping the Standard Errors
    - Bootstrapping the Standard Errors in Regressions
    - Bootstrapping the Critical Values and p Values

Introduction

- To characterize the systematic behavior of the variable of interest y based on the explanatory variable $\boldsymbol{x}$, we have learned how to fit the sample $(y_i, \boldsymbol{x}_i')'$, $i = 1, \ldots, n$, with a linear regression model. Yet, researchers choose linear model mainly because of its simplicity and convenience. More generally, we may consider fitting the sample with a nonlinear or nonparametric model f.
- Taking the mean squared error (MSE), $\mathbb{E}[y - f(\boldsymbol{x})]^2$, as the criterion for model fitness, we may "train" (estimate) a nonlinear function f by minimizing the sample counterpart of MSE:

$$\text{MSE}_{\text{training}} = \frac{1}{n} \sum_{i=1}^{n} [y_i - f(\boldsymbol{x}_i)]^2;$$

this is just the LS criterion when f is a linear function.

- To select a trained model $\hat{f}$ from a collection of models, it is common to compare their model performance based on (adjustment of) the training MSE, e.g., $\bar{R}^2$ in linear regression.
- Instead of using the same sample for model training and selection, a different approach is to evaluate model performance based on a previously unseen sample (test sample): $(\eta_i, \boldsymbol{\xi}_i)$, $i = 1, \ldots, m$. To this end, we compute the testing MSE of $\hat{f}$:

$$\text{MSE}_{\text{testing}} = \frac{1}{m} \sum_{i=1}^{m} [\eta_i - \hat{f}(\boldsymbol{\xi}_i)]^2,$$

  and select the model with the lowest testing MSE.
- Clearly, a model with a low training MSE does not necessarily have a low testing MSE, and the selected model would be different if different testing samples are used.

Resampling

- Despite that we have only a given sample, modern statistics introduces the concept of resampling. The idea is to draw "new" samples from the original sample and train the model using each of the "new" samples. This is in sharp contrast with classical statistics in which the sample is used for training only once.
- While resampling is computationally demanding, it enables us to extract more information from the sample and to assess the model performance in very different ways. As computation is now more powerful and less costly than before, computational simplicity is no longer a major concern in statistical inference.
- In this lecture we will discuss two important resampling methods: Cross Validation (CV) and Bootstrap.

The Validation Set Approach

- The Validation Set Approach randomly splits the given sample into two sub-samples of equal size, a training sample and a validation set as the test sample.
- The training sample is used only for training models, and the validation set is used only for model selection. Each trained model is applied to predict the outcomes in the validation set and results in a testing MSE. A model is selected among a collection of models when it has the lowest testing MSE.
- The trained model and its performance in the validation set depend on how the sample is partitioned; different partitions may lead to different selected models.

The Bias-Variance Trade-Off

Given the observation $(y_0, \boldsymbol{x}_0')'$, write $y_0 = f(\boldsymbol{x}_0) + \varepsilon$. It can be shown that the expected MSE at $\boldsymbol{x}_0$ can be decomposed into 3 parts:

$$
\begin{aligned}
&\mathbb{E}[y_0 - \hat{f}(\boldsymbol{x}_0)]^2 \\
&\quad = \mathbb{E}\big[\big(f(\boldsymbol{x}_0) - \mathbb{E}(\hat{f}(\boldsymbol{x}_0))\big) - \big(\hat{f}(\boldsymbol{x}_0) - \mathbb{E}(\hat{f}(\boldsymbol{x}_0))\big) + \varepsilon\big]^2 \\
&\quad = (\text{bias})^2 + \operatorname{var}(\hat{f}(\boldsymbol{x}_0)) + \operatorname{var}(\varepsilon),
\end{aligned}
$$

where $\operatorname{var}(\varepsilon)$ does not depend on $\hat{f}$ and is not reducible. In general, an $\hat{f}$ cannot achiever low bias and low variance at the same time; instead, there is usually a trade-off between the bias and variance. Intuitively, a more flexible model (method) fits training data more closely, and it may change considerably even when data change slightly. Thus, a more flexible model tends to have a smaller bias and a larger variance.

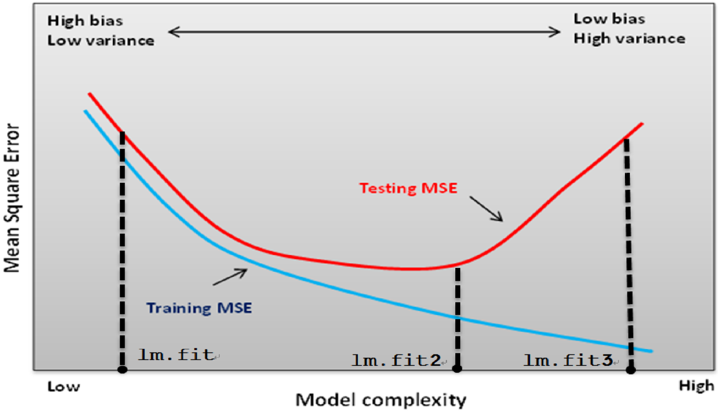
Example: Taiwan Traffic Data

The monthly traffic-related data in Taiwan from Feb. 2000 to July 2018 are taken from National Police Agency of the Ministry of Interior. We focus on the following variables:

- Death: The number of death due to traffic accidents in a month.
- AutoIncr: The number of increased automobiles in a month.
- MotorIncr: The number of increased motorcycles in a month.
- Time: Linear time trend.

As an example, we consider the following models:

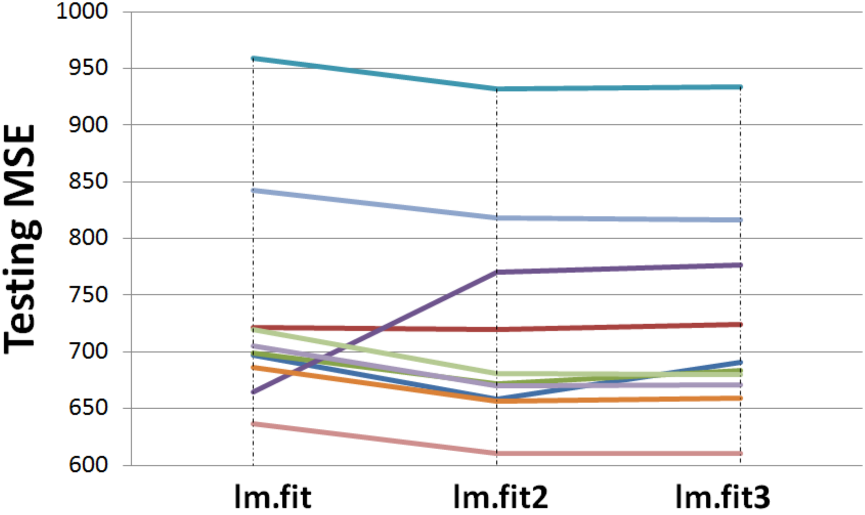
$$
\begin{aligned}
\texttt{lm.fit:} &\quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + u, \\
\texttt{lm.fit2:} &\quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + u, \\
\texttt{lm.fit3:} &\quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + \beta_3 \texttt{MotorIncr} + u.
\end{aligned}
$$



The estimated testing MSEs for `lm.fit`, `lm.fit2`, and `lm.fit3` are, respectively, 696.6768, 658.6364, and 690.9322.

Drawbacks of the Validation Set Approach

- The results of the Validation Set Approach depend on random partition of the sample and hence are quite arbitrary. To see this, we conduct 10 random partitions and plot the resulting testing MSEs in the next page. It turns out that the model with the lowest testing MSE may be any of the 3 competing models, depending on which partition is used.
- Another major drawback of the Validation Set Approach is that it does not fully utilize the sample information because only half of the sample is used for model training. This problem remains even when the sample is not partitioned equally.


Testing MSE
1000
950
900
850
800
750
700
650
600
lm.fit
lm.fit2
lm.fit3

Leave-One-Out Cross Validation

- Instead of using half of the sample as the validation set, we may take only one observation, say $(y_1, \boldsymbol{x}_1')$, for model validation and the remaining observations, $\{(y_2, \boldsymbol{x}_2'), ..., (y_n, \boldsymbol{x}_n')\}$, for model training. The resulting model is denoted as $\hat{f}_{-1}$. The estimate of the testing MSE is:

$$\mathsf{MSE}_1 = [y_1 - \hat{f}_{-1}(\boldsymbol{x}_1)]^2.$$

- For each $i = 2, \ldots, n$, take $(y_i, \boldsymbol{x}_i')$ as the validation set and the remaining observations for training $\hat{f}_{-i}$. We have the testing MSE:

$$\mathsf{MSE}_i = [y_i - \hat{f}_{-i}(\boldsymbol{x}_i)]^2, \quad i = 2, \ldots, n.$$

  Thus, every observation in the original sample will be "validated" using the model estimated from the remaining $n-1$ observations.

- The Leave-One-Out Cross Validation (LOOCV) estimate of the testing MSE is:

$$\mathrm{MSE}_{\mathrm{LOOCV}} = \frac{1}{n} \sum_{i=1}^{n} \mathrm{MSE}_i.$$

  A model is selected if it has the lowest $\mathrm{MSE}_{\mathrm{LOOCV}}$.
- Compared with the Validation Set Approach, LOOCV avoids random partition of the sample and utilizes almost the complete sample to train models. Yet, LOOCV is computationally demanding when n is large or when the model is difficult to train (e.g., a highly nonlinear model).
- The result below shows that, when the model f is linear and trained by minimizing MSE, the LOOCV estimate can be easily computed by doing LS regression once using the entire sample.

A Special Case of LOOCV: Linear Regression

Let $\hat{y}_i$ denote the fitted values of the linear model trained by OLS. Then,

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

where $h_i = \boldsymbol{x}_i'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{x}_i$, the i th diagonal term of $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$.

Proof: Given the data $\boldsymbol{y}$ $(n \times 1)$ and $\boldsymbol{X}$ $(n \times p)$, let $\boldsymbol{y}_{-i}$ and $\boldsymbol{X}_{-i}$ denote the sub-matrices of $\boldsymbol{y}$ and $\boldsymbol{X}$, each with the i th row deleted. The OLS estimators of regressing $\boldsymbol{y}$ on $\boldsymbol{X}$ and $\boldsymbol{y}_{-i}$ on $\boldsymbol{X}_{-i}$ are, respectively, $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$, and $\hat{\boldsymbol{\beta}}_{-i} = (\boldsymbol{X}'_{-i}\boldsymbol{X}_{-i})^{-1}\boldsymbol{X}'_{-i}\boldsymbol{y}_{-i}$. We can write $\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{X}'_{-i}\boldsymbol{y}_{-i} + \boldsymbol{x}_i y_i$ and

$$\boldsymbol{X}'\boldsymbol{X} = \sum_{j=1}^{n}\boldsymbol{x}_j\boldsymbol{x}_j' = \boldsymbol{X}'_{-i}\boldsymbol{X}_{-i} + \boldsymbol{x}_i\boldsymbol{x}_i'.$$

Writing $\boldsymbol{A} = \boldsymbol{X}'\boldsymbol{X}$, a well-known matrix inversion formula (the third equality) shows that:

$$\begin{aligned}
\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_{-i} &= \boldsymbol{x}_i'(\boldsymbol{X}'_{-i}\boldsymbol{X}_{-i})^{-1}\boldsymbol{X}'_{-i}\boldsymbol{y}_{-i} \\
&= \boldsymbol{x}_i'(\boldsymbol{X}'\boldsymbol{X} - \boldsymbol{x}_i\boldsymbol{x}_i')^{-1}(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i) \\
&= \boldsymbol{x}_i'\left(\boldsymbol{A}^{-1} + \frac{\boldsymbol{A}^{-1}\boldsymbol{x}_i\boldsymbol{x}_i'\boldsymbol{A}^{-1}}{1 - \boldsymbol{x}_i'\boldsymbol{A}^{-1}\boldsymbol{x}_i}\right)(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i) \\
&= \left(\boldsymbol{x}_i'\boldsymbol{A}^{-1} + \frac{h_i\boldsymbol{x}_i'\boldsymbol{A}^{-1}}{1 - h_i}\right)(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i),
\end{aligned}$$

where $h_i = \boldsymbol{x}_i'\boldsymbol{A}^{-1}\boldsymbol{x}_i$. As $\hat{y}_i = \boldsymbol{x}_i'\hat{\boldsymbol{\beta}} = \boldsymbol{x}_i'\boldsymbol{A}^{-1}\boldsymbol{X}'\boldsymbol{y}$, we have

$$\boldsymbol{x}_i'\hat{\boldsymbol{\beta}}_{-i} = \left(\frac{\boldsymbol{x}_i'\boldsymbol{A}^{-1}}{1 - h_i}\right)(\boldsymbol{X}'\boldsymbol{y} - \boldsymbol{x}_i y_i) = \frac{\hat{y}_i - h_i y_i}{1 - h_i}.$$

This is the prediction based on x_i and the training OLS estimator $\hat{\boldsymbol{\beta}}_{-1}$.

It follows that the prediction error of y_i is

$$y_i - \boldsymbol{x}_i' \hat{\boldsymbol{\beta}}_{-i} = \frac{y_i(1-h_i) - \hat{y}_i + h_i y_i}{1-h_i} = \frac{y_i - \hat{y}_i}{1-h_i},$$

and $\mathsf{MSE}_i == [(y_i - \hat{y}_i)/(1-h_i)]^2$, $i = 1, \ldots, n$. Consequently,

$$\mathsf{MSE}_{\mathsf{LOOCV}} = \frac{1}{n} \sum_{i=1}^{n} \mathsf{MSE}_i = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1-h_i} \right)^2 .$$

Remark: The i th diagonal element of $\boldsymbol{H}$ is $h_i = \boldsymbol{x}_i'(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{x}_i$, and hence

$$\sum_{i=1}^{n} h_i = \text{trace}(\boldsymbol{H}) = \text{trace}\boldsymbol{I}_p = p.$$

For simplicity, we may approximate h_i by p/n, so that $\mathsf{MSE}_{\mathsf{LOOCV}}$ is proportional to the sample MSE.

k-Fold Cross-Validation

- An approach simpler than LOOCV is k-Fold Cross-Validation: Randomly partition the sample into k groups of equal size and repeatedly take one group as a validation set and remaining data as the training sample. This leads to k estimates of the testing MSEs: $\text{MSE}_1, \ldots, \text{MSE}_k$. The resulting k-fold estimate of the testing MSE is:

$$\text{MSE}_{k\text{-fold}} = \frac{1}{k} \sum_{j=1}^{k} \text{MSE}_j.$$

- k-fold CV is computationally much simpler because it requires fitting only k ($k \ll n$) models. When $k = n$, this is just LOOCV; when $k = 2$, this amounts to implementing the Validation Set Approach twice by flipping the validation and training sets.

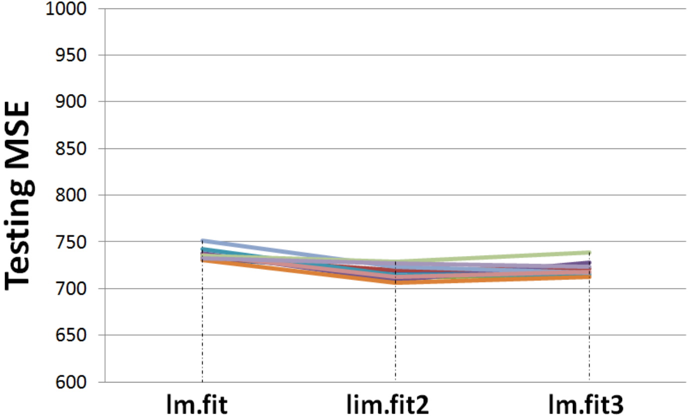
Example: Taiwan Traffic Data

Recall that we have three competing models in this example:

$$\begin{aligned}
\texttt{lm.fit:} &\quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + u, \\
\texttt{lm.fit2:} &\quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + u, \\
\texttt{lm.fit3:} &\quad \texttt{Death} = \beta_0 + \beta_1 \texttt{Time} + \beta_2 \texttt{AutoIncr} + \beta_3 \texttt{MotorIncr} + u.
\end{aligned}$$

- $\text{MSE}_{\text{LOOCV}}$ for `lm.fit`, `lm.fit2`, and `lm.fit3` are, respectively, 738.3419, 718.1488, and 718.6004,
- $\text{MSE}_{10\text{-fold}}$ for `lm.fit`, `lm.fit2`, and `lm.fit3` are, respectively, 733.5457, 719.6191, and 721.1696

Consider 10 different random partitions. The plot below shows that the resulting $\text{MSE}_{10\text{-fold}}$ are quite stable (MSE between 700 and 750), in contrast with those based on the Validation Set Approach.



Comparison between Different Methods

The discussion below is based on p. 205 of JWHT (2021).

- While LOOCV averages n fitted models that are based on highly positively correlated training samples, k-fold CV averages only k fitted models that are based on less correlated training samples (with less overlapping observations). As such, k-fold CV tends to have a smaller variance than does LOOCV.
- While the Validation Set Approach utilizes only half of the sample to train models, k-fold CV makes use of $(k-1)n/k$ observations for model training. Thus, k-fold CV yields a smaller bias than does the Validation Set Approach; note that LOOCV gives approximately unbiased estimate.
- Different choices of k lead to different bias-variance trade-off. In practice, it is common to set $k = 5$ or $k = 10$.

Comparison Based on Simulations

We conduct simulations to compare the performance of different methods. We generate the data x_i and ϵ_i as i.i.d. $\mathcal{N}(0,1)$ and

$$y_i = 1 - 2x_i + 1\,x_i^2 + \epsilon_i,$$

and train the following models with the sample of 10 million observations:

$$\begin{aligned}
\mathcal{M}_1 &: y_i = \beta_0 + \beta_1 x_i + u_i, \\
\mathcal{M}_2 &: y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + u_i, \\
\mathcal{M}_3 &: y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + u_i.
\end{aligned}$$

The "true" testing MSEs are computed from the test sample with 10 million observations: 3, 1, and 1.

Simulation Results

The simulations are based on the training sample of 100 observations with $5,000$ replications.

		LOOCV	10-Fold CV	5-Fold CV	Validation Set
$\widehat{\mathcal{M}}_1$	Bias	0.1418	0.1537	0.1754	0.2819
	Variance	0.8646	0.8815	0.8915	1.6860
$\widehat{\mathcal{M}}_2$	Bias	0.0306	0.0348	0.0427	0.0711
	Variance	0.0217	0.0223	0.0237	0.0539
$\widehat{\mathcal{M}}_3$	Bias	0.0595	0.0689	0.0810	0.1650
	Variance	0.0276	0.0298	0.0327	0.1887

The bias is smaller when k is larger: LOOCV (Validation Set Approach) has the smallest (largest) bias. Yet, we find the variance is also smaller when k is larger in this case: LOOCV (Validation Set Approach) has the smallest (largest) variance, cf. the discussion of JWHT (2021).

Bootstrap

- In conventional statistical analysis, we rely on a given sample to estimate unknown properties of the population and draw inference by testing the estimation results. These tests depend on strong assumptions on data (e.g. normality) or asymptotic approximation.
- The conventional approach may fail when a test statistic is difficult to construct or lack an analytic form, or when its asymptotic distribution is a poor approximation to the exact distribution.
- Efron (1979) introduces the idea of Bootstrap. This approach treats the original sample as the "population", from which many "new" samples can be drawn randomly. These sample information are then used to assess the performance of the original estimation results. As such, bootstrap utilizes the sample information more than once, in contrast with the conventional approach.

Bootstrapping the Standard Errors

Example: Given two assets with returns x and y (with means μ_x and μ_y, variances σ_x^2 and σ_y^2, and covariance σ_{xy}), these two assets can be allocated using the optimal fraction α_o that minimizes $\mathrm{var}(\alpha x + (1-\alpha)y)$:

$$\alpha_o = \frac{\sigma_y^2 - \sigma_{xy}}{\sigma_x^2 + \sigma_y^2 - 2\sigma_{xy}}.$$

Given the sample $\mathcal{S} = \{(y_i, x_i), i = 1, \ldots, n\}$, we can easily calculate the estimators $\hat{\sigma}_x^2$, $\hat{\sigma}_y^2$, $\hat{\sigma}_{xy}$, and hence

$$\hat{\alpha} = \frac{\hat{\sigma}_y^2 - \hat{\sigma}_{xy}}{\hat{\sigma}_x^2 + \hat{\sigma}_y^2 - 2\hat{\sigma}_{xy}}.$$

Yet, computing the standard error of $\hat{\alpha}$ is not straightforward.

Let $\mathcal{S} = \{(y_i, x_i), i = 1, \ldots, n\}$ denote the original sample. Below is the procedure for bootstrapping the standard error.

1. For each $b = 1, 2, \ldots, B$, randomly draw n observations from $\mathcal{S}$ with replacement and obtain the b th bootstrapped sample:
$$\mathcal{S}_b = \{(y^*_{b,1}, x^*_{b,1}), \ldots, (y^*_{b,n}, x^*_{b,n})\};$$
this is also known as case resampling.
2. For each bootstrapped sample $\mathcal{S}_b$, compute $\hat{\sigma}^*_{y,b}$, $\hat{\sigma}^*_{x,b}$ and $\hat{\sigma}^*_{xy,b}$ to obtain $\hat{\alpha}^*_b$, $b = 1, 2, \ldots, B$.
3. The resulting $\hat{\alpha}^*_1, \ldots, \hat{\alpha}^*_B$ constitute an empirical distribution of $\hat{\alpha}$, with the sample mean: $\bar{\hat{\alpha}}^* = B^{-1} \sum_{b=1}^{B} \hat{\alpha}^*_b$, and the standard error:
$$\text{se}^*(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(\hat{\alpha}^*_b - \bar{\hat{\alpha}}^*\right)^2}.$$

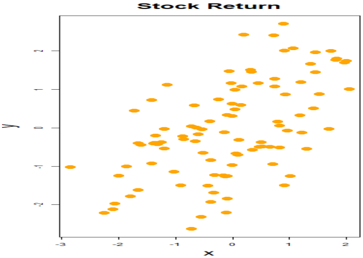
Remarks:

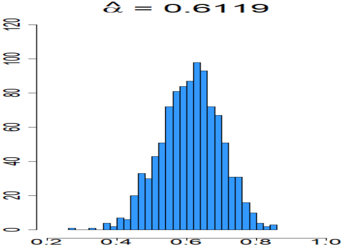
- In bootstrap, random sampling an i.i.d. sample creates another i.i.d. sample. As sampling is done with replacement, some observations in $\mathcal{S}$ may not be drawn, but some observations in $\mathcal{S}$ may be drawn more than once. Thus, bootstrapped samples are different in general.
- From the bootstrapped samples we obtain B bootstrapped statistics which form an empirical distribution of the statistic from the original sample. A large B thus leads to better approximation to the true distribution.
- Bootstrap is readily applicable to difficult estimation problems and does not require strong assumptions on data or model. It has been found that bootstrap usually leads to more reliable inferences than those based on asymptotic approximation.

Now suppose the return pairs $(y, x)'$ are generated according to:

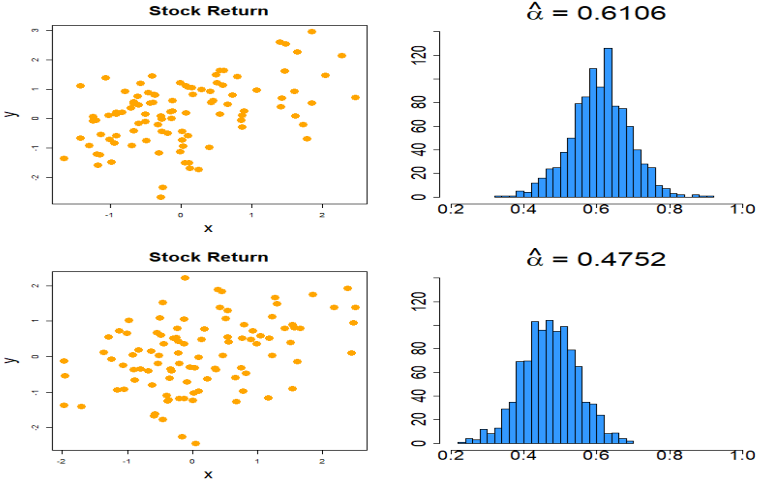
$$\begin{pmatrix} y \\ x \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.25 & 0.5 \\ 0.5 & 1 \end{pmatrix} \right).$$

Here, $\alpha_o = 0.6$. Below is the scatter plot of a random sample of 100 observations generated from this distribution (left), for which $\hat{\alpha} = 0.6119$, and the bootstrapped distribution of $\hat{\alpha}$ (right) with $\bar{\hat{\alpha}}^* = 0.6113$ and SE$^*(\hat{\alpha}) = 0.0853$. This is close to the "true" SE (0.081) which is calculated from $100,000$ random samples.





Below are the scatter plots of another 2 random samples for which $\hat{\alpha}$ are 0.6106 and 0.4752. Bootstrap then yields $\bar{\hat{\alpha}}^{*}$: 0.6089 and 0.4711, and $\mathrm{SE}^{*}(\hat{\alpha})$: 0.0806 and 0.0762, which are also close to the "true" SE (0.081), even when $\hat{\alpha}$ is far away from 0.6.



Standard Errors in Regressions

Given the sample $\mathcal{S} = \{(y_i, \boldsymbol{x}_i'), i = 1, \ldots, n\}$, regressing y_i on $\boldsymbol{x}_i$ yields

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_k x_{ik} + \hat{u}_i, \quad i = 1, \ldots, n,$$

where $\hat{\boldsymbol{\beta}}$ is the vector of the OLS estimates and $\hat{u}_i$ the residuals. We have learned that the standard error of the coefficient estimate: $\hat{\beta}_j$ is the square root of the $(j+1)$ th diagonal element of the classical estimator: $\hat{\sigma}^2(\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i')^{-1}$, or the Eicker-White-type estimator:

$$\left(\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1} \left(\sum_{i=1}^n \hat{u}_i^2 \boldsymbol{x}_i\boldsymbol{x}_i'\right) \left(\sum_{i=1}^n \boldsymbol{x}_i\boldsymbol{x}_i'\right)^{-1}.$$

Instead, we may compute standard errors via bootstrap.

Paired Bootstrap

Given the sample $\mathcal{S}$ of n observations, the procedure for bootstrapping the regression standard error is:

1. For each $b = 1, 2, \ldots, B$, randomly draw n observations from $\mathcal{S}$ with replacement and obtain the b th bootstrapped sample: $\mathcal{S}_b = \{(y^*_{b,i}, \boldsymbol{x}^{*\prime}_{b,i}), i = 1, \ldots, n\}$.
2. For each $\mathcal{S}_b$, regress $y^*_{b,i}$ on $\boldsymbol{x}^*_{b,i}$ to obtain $\hat{\boldsymbol{\beta}}^*_b$.
3. The bootstrapped standard error of $\hat{\beta}_j$ is:

$$\text{se}^*(\hat{\beta}_j) = \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \big(\hat{\beta}^*_{j,b} - \bar{\hat{\beta}}{}^*_j\big)^2},$$

where $\bar{\hat{\beta}}{}^*_j = B^{-1} \sum_{b=1}^{B} \hat{\beta}^*_{j,b}$

Residual Bootstrap

The residual bootstrap suggests bootstrapping the original residuals $\hat{u}_i = y_i - \boldsymbol{x}_i'\hat{\boldsymbol{\beta}}$ while keeping the regressors $\boldsymbol{x}_i$ fixed.

1. For each $b = 1, 2, \ldots, B$, randomly draw n observations from the residuals $\hat{u}_i$ with replacement, denoted as $\hat{u}_{b,i}^*$, and compute $y_{b,i}^*$ as

$$y_{b,i}^* = \boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + \hat{u}_{b,i}^*.$$

The bth bootstrapped sample is $\mathcal{S}_b = \{(y_{b,i}^*, \boldsymbol{x}_i'), i = 1, \ldots, n\}$.

2. For each $\mathcal{S}_b$, regress $y_{b,i}^*$ on $\boldsymbol{x}_i$ to obtain $\hat{\boldsymbol{\beta}}_b^*$.
3. The bootstrapped standard error of $\hat{\beta}_j$ is:

$$\text{se}^*(\hat{\beta}_j) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\beta}_{j,b}^* - \bar{\hat{\beta}}_j^*)^2}.$$

Example

- Generate x_i as i.i.d. $\mathcal{N}(0,1)$, u_i as i.i.d. $\mathcal{N}(0,1)$ or $t(4)$, $i = 1, \ldots, 30$. Then, generate y_i according to:

  $$y_i = \beta_0 + \beta_1 x_i + u_i, \quad \beta_0 = 2, \beta_1 = 4.$$

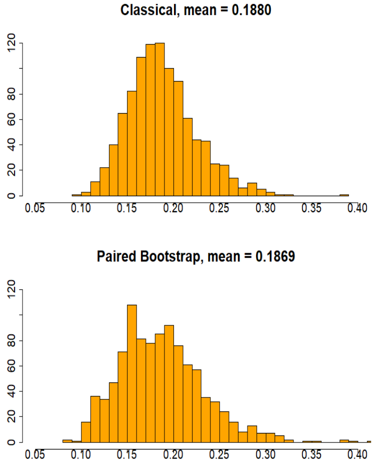
  This gives the sample $\mathcal{S} = \{(y_i, x_i), i = 1, \ldots, 30\}$.

- Regress y_i on $\boldsymbol{x}_i = (1, x_i)'$ to obtain $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)'$. The standard error of $\hat{\beta}_1$ is the square root of the 2nd diagonal element of the classical estimator: $\hat{\sigma}^2(\sum_{i=1}^{30} \boldsymbol{x}_i \boldsymbol{x}_i')^{-1}$, or the Eicker-White-type estimator:

  $$\left(\sum_{i=1}^{30} \boldsymbol{x}_i \boldsymbol{x}_i'\right)^{-1} \left(\sum_{i=1}^{30} \hat{u}_i^2 \boldsymbol{x}_i \boldsymbol{x}_i'\right) \left(\sum_{i=1}^{30} \boldsymbol{x}_i \boldsymbol{x}_i'\right)^{-1}.$$

Conventional/Bootstrapped Standard Errors: $u_i \sim \mathcal{N}(0,1)$

Below are the simulated distributions of the SEs, based on $1,000$ simulated samples of 30 observations. Note that the "true" SE is 0.1922.



Classical, mean = 0.1880

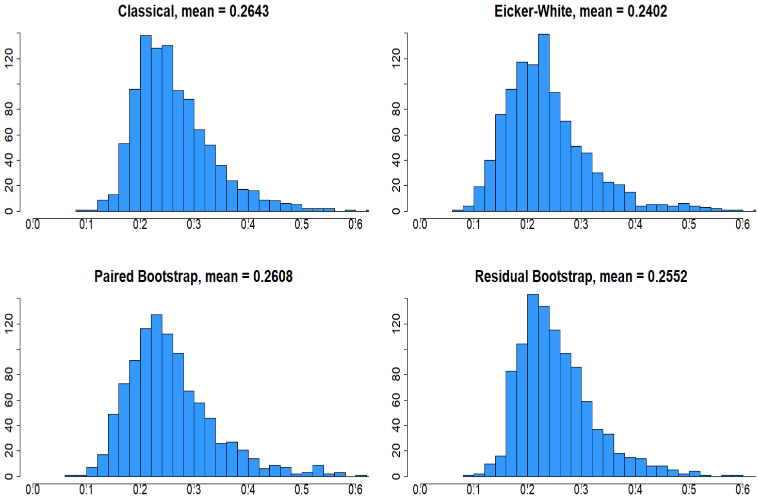
Eicker-White, mean = 0.1727

Paired Bootstrap, mean = 0.1869

Residual Bootstrap, mean = 0.1814

Conventional/Bootstrapped Standard Errors: $u_i \sim t(4)$

Note that the “true” SE is 0.2724.



Bootstrapping the Critical Values

For hypothesis testing, we may bootstrap the test statistic and its critical values. Consider the example of the t statistic: $t_j = (\hat{\beta}_j - c)/\mathrm{se}(\hat{\beta}_j)$, with c the hypothetical value.

- Compute bootstrapped $\hat{\beta}^*_{j,b}$ and $\mathrm{se}^*(\hat{\beta}_j)$ using the paired or residual bootstrap. The resulting bootstrapped t_j is:

  $$t^*_{j,b} = (\hat{\beta}^*_{j,b} - \hat{\beta}_j)/\mathrm{se}^*(\hat{\beta}_j), \quad b = 1, 2, \ldots, B.$$

  Note that $t^*_{j,b}$ is centered at $\hat{\beta}_j$, rather than c.

- Order $t^*_{j,b}$ in an ascending order. For the significance level α, the critical value of the one-sided test is the $(1-\alpha)$ th quantile of $t^*_{j,b}$, and that of the two-sided test are the $(\alpha/2)$ th and $(1-\alpha/2)$ th quantiles.

The t_j statistic is then compared with the bootstrapped critical values.

Bootstrapping the p Values

Similarly, we may bootstrap the p value of the t_j statistic. To this end, we bootstrap t_j to obtain

$$t^*_{j,b} = (\hat{\beta}^*_{j,b} - \hat{\beta}_j)/\mathsf{se}^*(\hat{\beta}_j), \quad b = 1, 2, \ldots, B.$$

The bootstrapped p value of t_j is the proportion of $t^*_{j,b}$ greater than t_j (i.e., tail probability of the empirical distribution based on $t^*_{j,b}$):

$$p^* = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}(t^*_{j,b} > t_j),$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Comparing the Paired and Residual Bootstraps

Letting $\tilde{u}^*_{b,i} = y^*_{b,i} - \boldsymbol{x}^{*\prime}_{b,i}\hat{\boldsymbol{\beta}}$, the OLS estimator from the paired bootstrap can be written as:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^*_b &= \left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}\boldsymbol{x}^{*\prime}_{b,i}\right)^{-1}\left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}y^*_{b,i}\right)\\ &= \hat{\boldsymbol{\beta}} + \left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}\boldsymbol{x}^{*\prime}_{b,i}\right)^{-1}\left(\sum_{i=1}^n \boldsymbol{x}^*_{b,i}\tilde{u}^*_{b,i}\right).\end{aligned}$$

As y^*_i and $\boldsymbol{x}^*_i$ are drawn jointly in the paired bootstrap, there is "simultaneity" problem in the regression of $y^*_{b,i}$ on $\boldsymbol{x}^*_{b,i}$ ($\boldsymbol{x}^*_{b,i}$ and $\tilde{u}^*_{b,i}$ are correlated). As such, $\hat{\boldsymbol{\beta}}^*_b$ may differ from $\hat{\boldsymbol{\beta}}$ substantially, and their difference does not vanish in the limit. This affects the bootstrapped critical values because the bootstrapped statistics are centered at $\hat{\boldsymbol{\beta}}$.

The residual bootstrap avoids the aforementioned problem by fixing $\boldsymbol{x}_i$ and bootstrapping the residuals $\hat{u}_i$. Then, the bootstrapped residuals are $\hat{u}^*_{b,i}$, and $y^*_{b,i}$ are computed as $\boldsymbol{x}_i'\hat{\boldsymbol{\beta}} + \hat{u}^*_{b,i}$. Regressing $y^*_{b,i}$ on $\boldsymbol{x}_i$ we obtain the OLS estimator:

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^*_b &= \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{x}_i y^*_{b,i} \right) \\
&= \hat{\boldsymbol{\beta}} + \left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i' \right)^{-1} \left(\sum_{i=1}^{n} \boldsymbol{x}_i \hat{u}^*_{b,i} \right),
\end{aligned}$$

where $\boldsymbol{x}_i$ and $\hat{u}^*_{b,i}$ are now uncorrelated so that the second term on the right-hand side vanishes in the limit.

References and Acknowledgement

References

1. James, G., D. Witten, T. Hastie, and R. Tibshirani (2021). *An Introduction to Statistical Learning, with Applications in R*, 2nd edition, New York: Springer. (JWHT (2021))
2. Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*, Second Edition, New York: Springer.

Some of the figures in this presentation are taken from JWHT (2021) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani