- 이번주 진도가 4,5장입니다. 발제는 30분내로 두개를 다 할것이구, issue는 하나만 만들겟습니당
- 작성자 : 11기 고동영
- 발제용이 아닌 원본 : https://godongyoung.github.io/%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D/2018/01/26/ISL-Resampling-Methods_ch5.html

재표본(Resampling)은 통계학에서 빼놓을 수 없는 요소이다. 간단히 말하자면 이는 training set에서 반복해서 sample을 뽑고, 거기에 반복해서 model을 적합시켜보는 것이다. 이는 기존의 training set **전체를 단지 한번만 쓰는것** 보다 더 **추가적인 정보**(어떤것이든! 생각보다 많다)를 줄 수 있는데, 예를들면 다음과 같다.

하나의 training set에 대해 수많은 sample들을 뽑아보고, 거기에 각각 선형회귀적합을 해본다. 이를 통해 기존에 한번만 적합시켰을때는 할 수 없었던, 우리의 model이 데이터가 달라짐에 따라 어느정도 **범위**에 있을 것인지에 대한 평가를 해볼 수 있게 된다.

여기에선 가장 많이 쓰이는 resampling method인 Cross-validation과 bootstrap을 다룰 것이다.

> 여기 안에 들어간것은 개인적인 참고를 위한 지엽적인 부분이니, 안읽으셔도 됩니다.

5.1 Cross-Validation

앞절에서 다루었듯이, test error 와 training error는 다르다. 모델을 적합시키는데 쓰이지 않은, 즉 주어지지 않은 data(test set)에 대한 오류율이 test error인데, 이걸 낮추는게 모델의 최종목적이라 할 수 있다. 반면 training error는 주어진 데이터에 대한 오류율로, 특히나 overfitting의 경우 실제 error율을 과소평가할 수도 있다.

평가를 하기 위한 test set이 따로 주어져 있다면 좋겠지만, 현실에선 그렇지 않은 경우가 훨씬 많다. (내일의 주가를 예측하는 문제를 생각해보자.) 이를 해결하기 위해 2가지 방법이 쓰이는데, **1)** training error rate에 수학적인 보정을 가하여 test error를 간접적으로 추정하는것 (6장에서 다룸) **2)** training set중 몇개를 따로 빼내서(**hold out**) test error를 **직접적으로 추정**하는 방법. 여기선 후자의 방법을 다룰 것이다.

5.1-1 The Validation set Approach

Validation set approach는 test error을 추정하는 가장 간단한 방법인데, 간단하게 우리의 training data를 **random**하게 **반** 잘라서, test error 추정을 위해 따로 빼놓는 것이다.(그냥 반이라고 생각하는 경우가 많은데, random하게 잘라야된다. 혹시나 순서에 따른 패턴이 있을수도 있으니!)

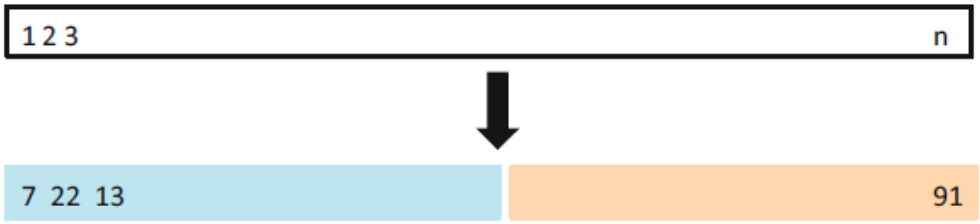
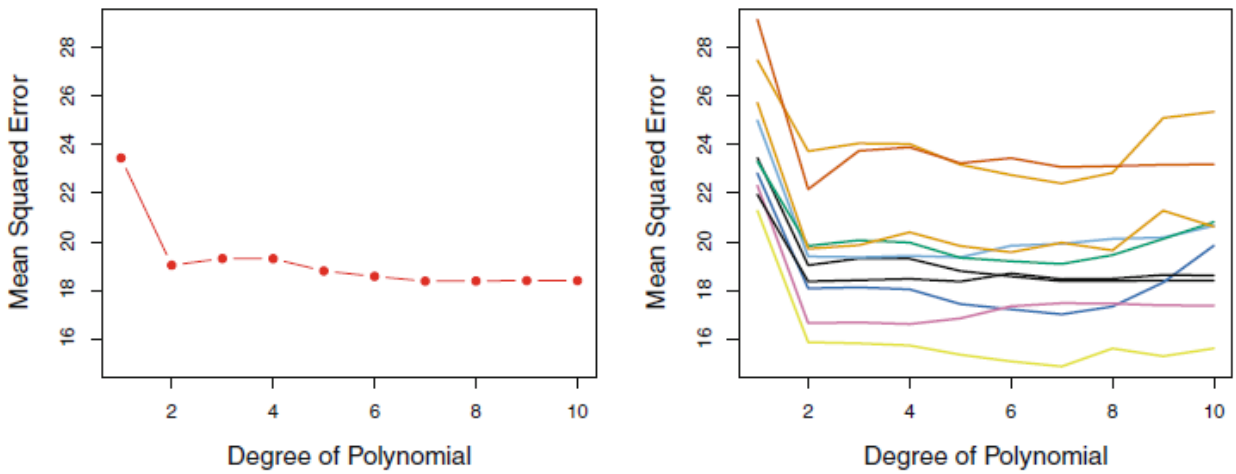


FIGURE 5.1. *A schematic display of the validation set approach. A set of n observations are randomly split into a training set (shown in blue, containing observations 7, 22, and 13, among others) and a validation set (shown in beige, and containing observation 91, among others). The statistical learning method is fit on the training set, and its performance is evaluated on the validation set.*

training set에서 떨어져 나온 이 data set들을 **validation set**이라고 부른다. (남은) training set에서 적합을 시키고 이를 validation set에 대해 error rate(회귀문제에선 대표적으로 MSE로 구한다)를 구하여 test error rate를 추정하는 것이다.

예를 들어 3장의 회귀분석에서, 자료가 U자형의 띄어 선형적합이 잘 안되는것 같아 다항회귀를 고려하고자 한다. 이때, 어떤 모델(2차항 모델, 3차항모델 등등)이 실제 관계에 잘 적합하는지를 알고 싶다. 그 경우 주어져있던 자료가 392개 라면 이를 196개로 나누어, 하나는 training set으로 적합에 사용하고, 남은 196개의 validation set의 자료를 통해 error rate를 추정해보는 것이다. (선형회귀에서는 계수의 p-value로 이를 알 수 있지만, 복잡한 모델일수록 이를 구하기 힘들다) 쉽게 말해, 모의고사용 문제를 따로 빼놓아, 평가를 해본다고 생각하면 된다.

그러나 이 방법은 랜덤하게 **'반이나'** 자른다는 점에서, 그 단점이 있다. data set이 어떻게 잘리느냐에 따라서 model의 변동이 심하고, 그에 따라 test MSE의 추정치도 심하게 변화하기 때문이다. 여러 방법에 따른 test MSE를 추정하고자 하였던 원 목적을 생각해보았을때, 이는 좋은 결과가 아니다. (은근히 많이 헷갈려한다. 우리는 model 자체를 평가하려는게 아니다. 참고)



위의 U자형 자료에 대한 회귀 예시로 돌아와, 왼쪽은 validation set approach를 한번만 해본것, 오른쪽은 10번 반으로 잘라서, 10번 해본것이다. 둘다 2차항 이상으로 적합하는 것에 큰 효과가 없음을 알려주긴 한다. 그러나, **1)** model에 따라 (어떻게 반으로 잘렸는지에 따라) **추정된 test MSE의 변동이 천차만별**이고, 심지어 **최적의 validation MSE를 위한 차수**도 천차만별이다. (고로 어떤 차수를 쓸것인지 불분명해진다) 또한, **2)** 자료의 수를 반이나 줄였다는 점에서, **추정의 성능역시 떨어지게** 된다. 반개의 자료만으로 적합한 모델은, 전체 자료로 적합하였을 경우 모델이 가졌을 test MSE보다 더 클 수 밖에 없다.(덜 정확할테니까) 즉, test error rate를 overestimate하게 된다.

> 뒤에서 나오겠지만, 사실 validation을 하기 위해 전체 자료 n개 중 일부를 빼는 순간, 전체 데이터(n개)로 할 수 있는 적합의 test error rate에 대한 추정은 아주 약간이라도 overestimate될 수 밖에 없다. 1개라도 더 작은 데이터셋을 가지고 분석을 한것이니까.

5.1-2 Leave-One-Out Cross-Validation

Leave-One-Out Cross-Validation, 줄여서 LOOCV는 이러한 validation set approach의 단점을 줄이고자 하는 방법이다. 이 방법은 validation set으로 반을 잘라내는 것이 아니라, **한개**만 따로 빼낸다. 그리고 전체 자료n개중 나머지 n-1개의 training set으로 적합을 한뒤, **하나의 자료에 대해서** error rate를 계산한다. ($MSE_1 = \frac{y_1 - \hat{y}_1}{1}$) 그리고, 위의 방법을 **모든 자료**에 **반복**한다.

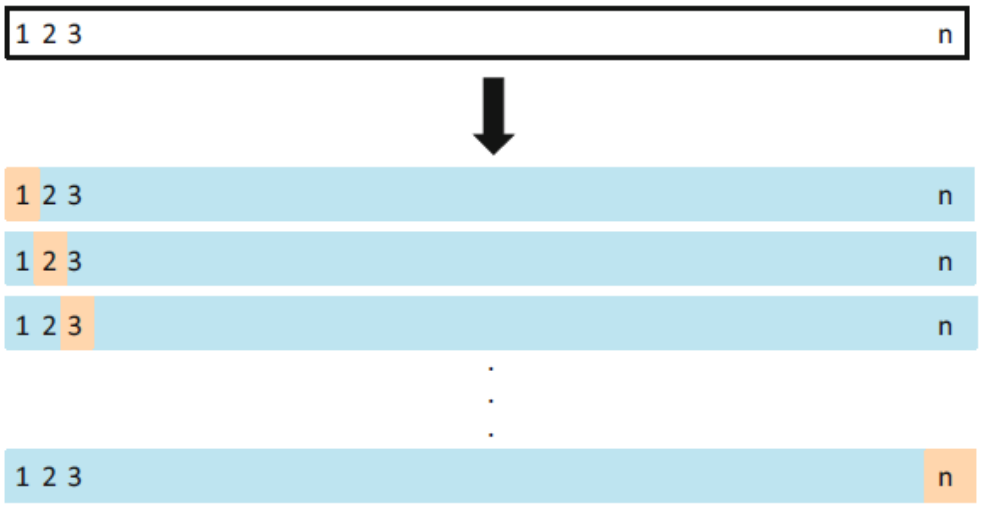


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

즉, 모든 자료n을 각각 한번씩 빼고, 각각에 경우 남은 n-1개의 training set 에대해 전부 적합을 한다. 이 경우 각 **추정된 n개의 test MSE**가 생길 것이고, 이를 최종적으로 **평균** 내주어, test error를 추정한다. 식으로 나타내면 다음과 같다.

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

사실, 이렇게 나온 LOOCV는 평균을 내기 전에도 모든 n번의 결과에 거의 차이가 없다. (이는 사실 뒤에도 나오지만, 우리의 data에 거의 최대로 맞추어 적합했기 때문. 즉 high variance를 의미한다. 이에 관해서는 뒤에서 더 설명한다)

LOOCV는, 다음과 같은 장점이 있다. **1)** 데이터의 반만을 가지고 적합을 하는 validation set approach와 달리, n-1개를 가지고 적합을 하기에 전체 training data의 test MSE에 대해 할 수 있는 거의 가장 정확한 추정을 할 수 있다. 즉, 거의 overestimate 하지 않는다. 바꿔말하면, **bias가 매우 적다**. **2)** 어떻게 training/validation을 나누느냐에 따라 결과가 달라졌던 validation set approach에 비해, LOOCV는 n번의 모든 경우를 자르고 평균을 내기에, **결과가 달라지지 않는다.**

데이터를 하나씩 빼서 n번의 적합을 하고 error를 각각 구하는것은 매우 소모적인 일일 수 있다. 그러나 least square로 적합한 선형회귀, 혹은 다항회귀에선, 이전 장에서 정의되었던 leverage statistic을 이용하여 더 빠르게 구할 수도 있다.

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

기존의 MSE를 구하는 식에서 분모에 leverage statistic h_i를 포함한 항이 추가된 식이다.

> leverage statistic은 least square에서 회귀계수들을 구하기 위한 정규방정식을 푸는 과정에서 자연스레 나오게 된다. 즉 least square로 푼 회귀문제에선 LOOCV는 전혀 어려운 문제가 아니다.

5.1-3 k-Fold Cross-Validation

너무 단순한 validation set approach와 너무 손이 많이 가는 LOOCV. 그들의 중간점이 바로 이 k-fold CV(Cross-Validation)이다. 전체 데이터를 특정한 수 k개(분석자가 정한다.)의 그룹으로 **random**하게 나눈다. 그리고 첫번째 그룹을 빼고, 남은 k-1개의 그룹으로 모델을 적합시키는 것이다. LOOCV에서 처럼, 이를 k번 반복한다. (첫번째 그룹을 빼서 MSE_1구하고, ..., k번째 그룹을 빼서 MSE_k 구하고.)

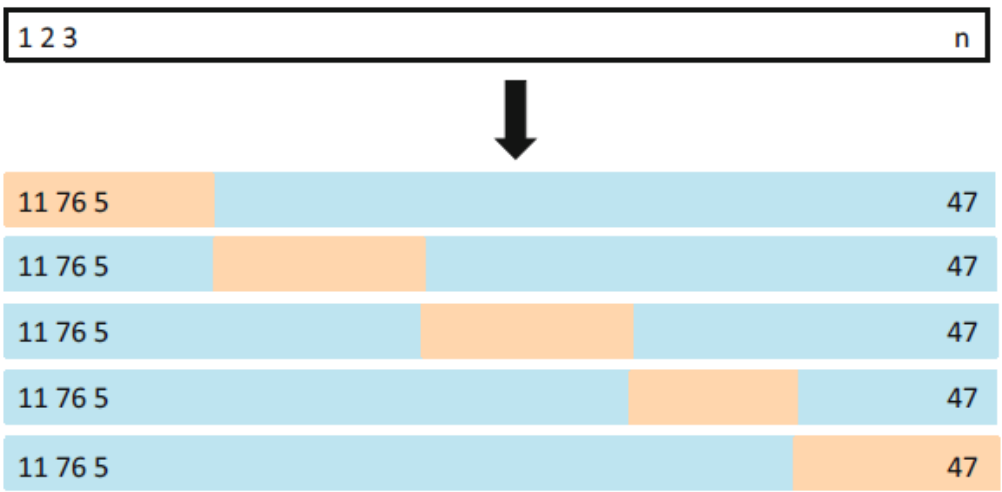


FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE estimates.*

최종적으로 나온 **k개의 MSE를 평균**을 내어 test MSE를 추정한다. 식으로는 다음과 같다.

$$\mathrm{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\mathrm{MSE}_i.$$

식을 보면 알겠지만, k=n이 되면 LOOCV이다. 실제에서는 k는 주로, 5개, 혹은 10개로 쓰인다. k=10이더라도, 이는 전체 데이터갯수 n번만큼의 적합을 해야했던 LOOCV에 비해 훨씬 수월한 방법이다.(10만개의 데이터셋이 있다 생각해보자) 그럼에도, 아래의 그림에서 볼 수 있듯이, 어떻게 10개의 그룹으로 나누는지에 따라서도 **추정이 크게 변동하지 않는다**. 왼쪽의 그림은 LOOCV, 오른쪽은 여러번 다르게 잘라본 10-fold CV이다.

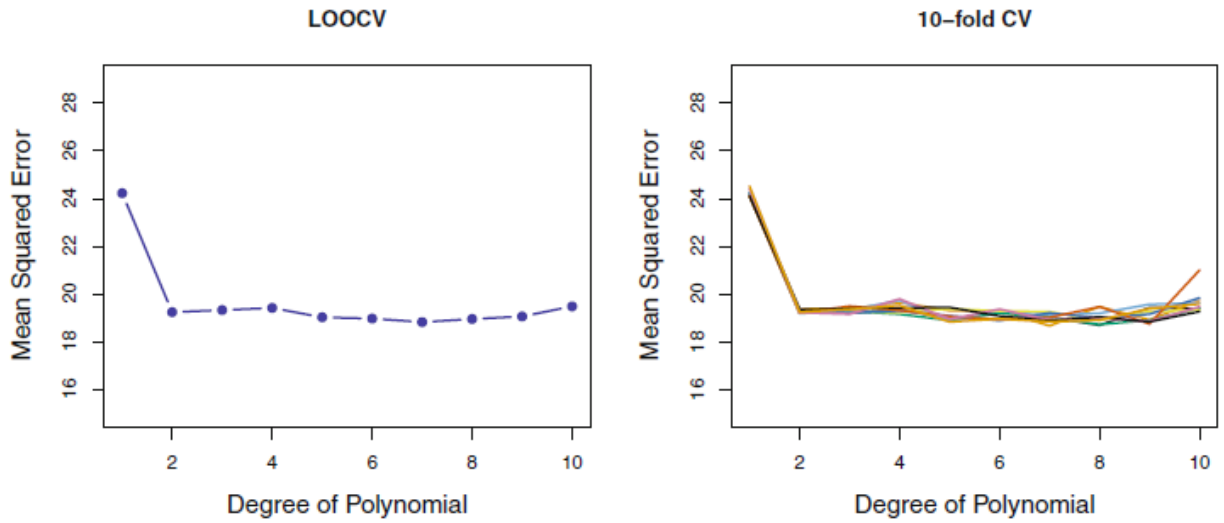


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10-*fold CV was run nine separate times, each with a different random split of the data into ten parts. The figure shows the nine slightly different CV error curves.*

이렇게 나온 k-fold CV는 LOOCV와 **성능**에서도 큰 차이를 보이진 않는다.

추가적으로 뒤에서 다루겠지만, computational 문제 외에도, **bias-variance trade-off의 측면**에서 LOOCV보다 강점이 있다.

5.1-4 Bias-Variance Trade-Off for k-Fold Cross-Validation

LOOCV와 k-fold CV의 관계에서도, **bias-variance trade-off**가 등장한다.

데이터의 반($\frac{n}{2}$개)만을 적합에 사용하는 validation set approach에서는, 그 수가 절대적으로 줄어 test error rate를 제대로 추정하지 못할 것(overestimate)이라는 것을 언급했었다. 사실 추정의 bias는 전체 full data를 쓰지 않겠다는 것에서 부터, 이미 전체 자료로 적합한 모델에 대한 test MSE에는 bias가 생길 수 밖에 없다.(참고. 추천) 이러한 점에서, **LOOCV는** 거의 전체 데이터은 n-1개의 데이터를 가지고 적합을 하기에 **근사적으로** unbiased 한 추정을 할 수 있다. 즉, **bias가 매우 낮다**.

그러나, 2장에서 다루었듯이 모델의 test MSE는 bias만으로 결정되지 않는다. ($Avg(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon)$ 상기.) 모델의 Variance가 얼마나 적은지도 역시 중요한데, **Variance 의 관점**에서 LOOCV는 **k-fold CV보다 못하다**.

LOOCV는 낮은 bias를 가지고 있는 반면 높은 variance를 가지고 있다. LOOCV에서 우리가 평균을 취해주는 n개의 적합된 모델들은, 1개의 데이터만을 빼고 적합을 했다는 점에서 거의 동일한 데이터(n-1개 중 n-2개가 동일)를 가지고 적합을 시킨, **거의 동일한 모델**이다. 바꿔말하면, 그 n개의 적합된 모델들은, 서로간에 **높은정도로 correlated 된** 모델들이다. 따라서 이 모델들을 합하고 나눈 LOOCV는 높은 Variance를 가지고 있는 것이다. (이는 기초 통계의 수식에서 알 수 있다. $Var(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} \sum_{j=1}^{n} Cov(X_i, X_j)$ 고로 'highly correlated 된 것들의 평균'의 분산은 'correlated 되지 않은 것들의 평균'의 분산보다 더 크다.)

더 쉽게 말해보자면, 만약 모집단에서 여러개의 training data를 뽑을 수 있다면 그때마다의 LOOCV의 test MSE 추정치는 변동할 것(high Variance)이라는 것이다. overfitting과 비슷하게 생각하면 이해하기 쉽다.

사실 validation set approach를 제외하고는 모든 CV방법은 서로 어느정도 겹치는 데이터를 가지고 적합을 하게 되고, 그에 따라 추정량의 Variance가 크게 된다. 다만 k=n인 경우, 즉 LOOCV에서 이 correlated가 최대가 된다. 그러나 k-fold의 경우, 각 sample간에 correlated 된 정도가 상대적으로 덜하다. 고로, 조금 **덜 겹치는**, k-fold CV가 Variance의 측면에서는 LOOCV보다 더 나은 방법이 된다. 쉽게말하자면, variance와 bias의 적당한 타협을 본 방안인 것이다. k가 몇인지에 따라 이 정도도 달라지지만, 주로 k=5, 혹은 10을 사용한다.

> 그러나 전체 데이터 셋이 극단적을 으로 작을 경우 데이터 자체의 noise가 심할 수 있으므로, 이 경우는 LOOCV를 쓰기도 한다. 이 경우는 k-fold도 (높은 bias를 가지고 있으면서도) 높은 variance를 가질 수 있기 때문

5.1-5 Cross-Validation on Classification Problems

이 전까지는 양적변수의 상황을 가정하고 MSE를 사용하는 경우에 대한 Cross-Validation을 논하였지만, 이는 질적변수에도 적용할 수 있다. 이 경우, 2장에서 논의 했던 대로 error rate를 측정하는 방식이 조금 달라질 뿐 모든 논의는 동일하다. 구체적으로 LOOCV의 경우 error rate는 다음과 같다.

$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq y_i)$$

예를 들어, 특정 데이터에 다중로지스틱회귀를 하는 것을 생각해보자. 몇차항 적합을 해야 가장 test MSE가 낮을 것인가에 대해 10-fold CV가 역시 좋은 추정을 해줄 수 있다.

5.2 The Bootstrap

Bootstrap은 실제로는 계산하기 어려운 추정량들의 불확실성(uncertainty. 그것이 어떤 통계량의 분산이던, 평균이던 뭐던)을 계산하는데 널리 쓰이는 강력한 통계기법이다. 선형회귀분석에서 계수의 분산같은 경우 수식적으로 표준편차를 구하는 것이 가능하지만, 많은 경우, 사실 정말 많은 경우에 수식적으로 통계량의 특성은 **정확히 구하지 못한다**. 이 경우, 이 bootstrap이 매우 강력한 툴로써 활용된다.

예를 들어 미지의 변수 X와 Y만큼의 투자이익을 주는 두 회사에 각각 α와 $1-\alpha$만큼의 비율을 투자한다 할 때, 전체 risk가 적은, 즉 $Var(\alpha X + (1-\alpha)Y)$를 최소화할 수 있는 방향으로 두 회사에 투자하려한다. 이 경우, 간단한 정리를 통해 해당 Var를 최소화하는 비율α는 다음과 같다는 것을 도출할 수 있다.

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

> $Var(\alpha X + (1-\alpha)Y) = \alpha^2 Var(X) + (1-\alpha)^2 Var(Y) + \alpha(1-\alpha)Cov(X,Y) = f(x)$라 두고
>
> $argmin_\alpha[\frac{df(x)}{d\alpha}]$하는 α를 구하면 된다. 나머지는 단순 전개에 정리하고 식 넘기는것.

여기서 $\sigma_X^2 = Var(X)$이고, $\sigma_{XY} = Cov(X,Y)$를 의미한다. 그러나 실제에선, 이를 최소화하는 α의 식을 구성하는 $Var(X), Var(Y), Cov(X,Y)$를 알 수 없다. 따라서 이를 가지고 있는 자료를 통해 추정할 수 밖에 없다. 이 경우 식은 다음과 같이 된다.

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}.$$

simulated 된 자료, 즉 만든 자료이므로, 여러번 sample을 뽑아 여러번 추정을 해보았더니, 추정된 $\hat{\alpha}$은 0.532~0.657에서 왓다갔다 했다. 그럼, 우리의 추정의 정확도, 즉 추정량 $\hat{\alpha}$의 standard deviation(표준편차)에 대해 알고싶다. 즉, 데이터를 **많이 뽑아 볼 수 있다면**, 그때마다 $\hat{\alpha}$는 얼마나 변동하는지를 알아보고 싶은 것이다.

회귀분석과 다르게 어떠한 분포 가정하에서 해당 값들을 구한것도 아니고 수많은 변수들이 복잡하게 얽혀있기에, 우리는 $\hat{\alpha}$의 sd에 대해서 알길이 없다. (위의 식을 알고 있어도 분산은 구할 수 없다.)그러나 이것은 simulated 된 데이터이므로, 이를 단순하게 100개쌍의 데이터를 1000번뽑아서, 어느정도 값이 나오는지, 얼마나 변동했는지 가늠해볼수가 있다.(!) 참고로 이때, 데이터의 true 값들은 다음과 같았다. $\sigma_x^2 = 1, \sigma_Y^2 = 1.25, \sigma_{XY} = 0.5$

1000번의 반복을 통해 만들어진 값들은 다음과 같다.

$$\bar{\alpha} = \frac{1}{1,000} \sum_{r=1}^{1,000} \hat{\alpha}_r = 0.5996,$$

$$\sqrt{\frac{1}{1,000 - 1} \sum_{r=1}^{1,000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083.$$

참값을 알고 있는 시점에서, 추정값들의 평균 $\bar{\alpha}$는 true값 0.6에 **매우 근사하다**. sd의 경우는 참값을 알 수 없지만, 이러한 방식으로 대략 0.08정도라고, 말할 수 있게 되었다.

이와 같이 수식으로는 계산하기 힘들더라도 실제로 **수없이 많이 뽑아볼 수 있다면** 그 참값에 매우 근사한 수치를 얻을 수 있다. 그러나 현실에서는, 위와 같이 데이터를 수없이 많이 뽑아볼 수가 없다. 그러나, Bootstrap은 수많은 **새로운 sample들**을 뽑아내어, 위와 같은 **추론을 가능하게** 한다.

어떤식으로 이게 가능할까? Bootstrap은, (위의 simulated data에서 했듯이) 모집단에서 샘플을 계속 새로 뽑는게 아니라, **가지고 있는 원래의 data set**에서 **복원추출로** 새로운 샘플을 계속 뽑는다. 그림으로 더욱 직관적인 이해가 가능하다.

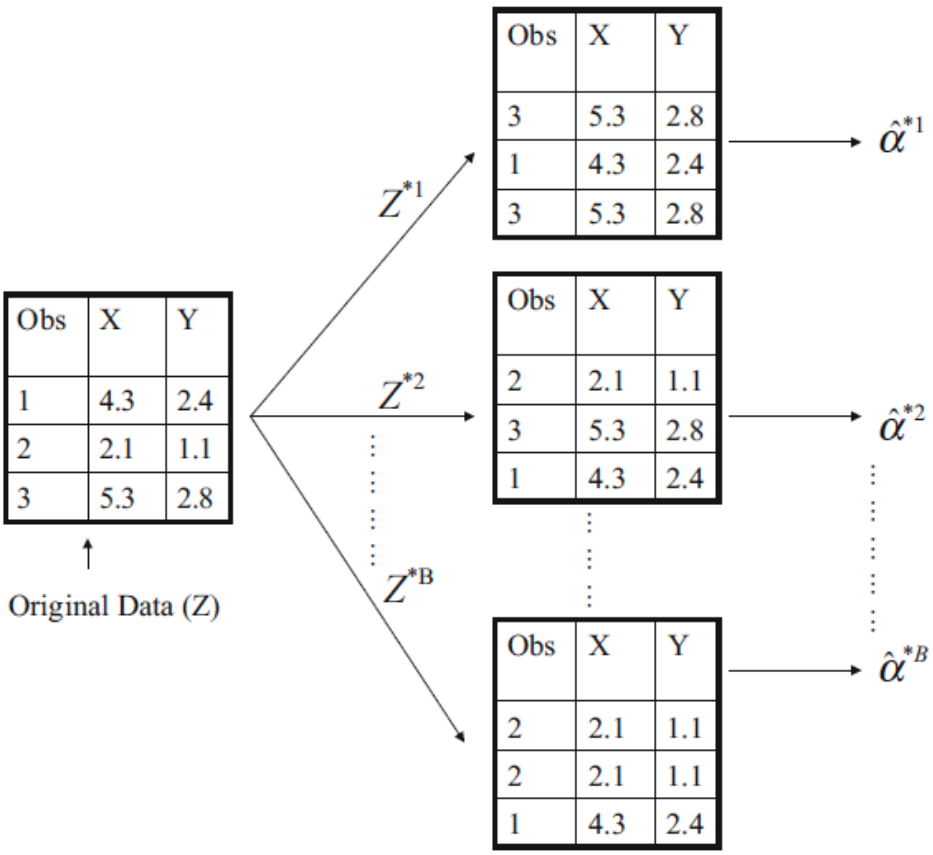


FIGURE 5.11. *A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.*

원래의 data Z는 3개밖에 없지만, 이를 가지고 복원추출을 반복 한다(이 경우 3개의 데이터로 구한 추정량에 대한 특성을 알고싶은 거기에, 3개 복원추출을 한다!). 그에따라 3번째 자료가 2번뽑힌 sample, 2번째 자료가 2번뽑힌 sample, 등등 수많은 **새로운 sample**이 생긴다. 이를 충분히 큰 수 B번을 반복하여 복원추출하면, 그에 따라 **B개의 sample**이 생기고, 그에 따라 **B개의 추정량** $\hat{\alpha}$가 생기게 된다. 이제 이 B개의 추정량 $\hat{\alpha}$을 가지고, 다음과 같이 SE를 구할 수 있다.

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}\left(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'}\right)^2}.$$

이에 대한 성능 확인을 위해 앞의 두 회사에 대한 투자의 예시에서, 100개의 data set 한개만을 가지고 bootstrap으로 1000개의 sample을 만들어 추정을 해보았다.

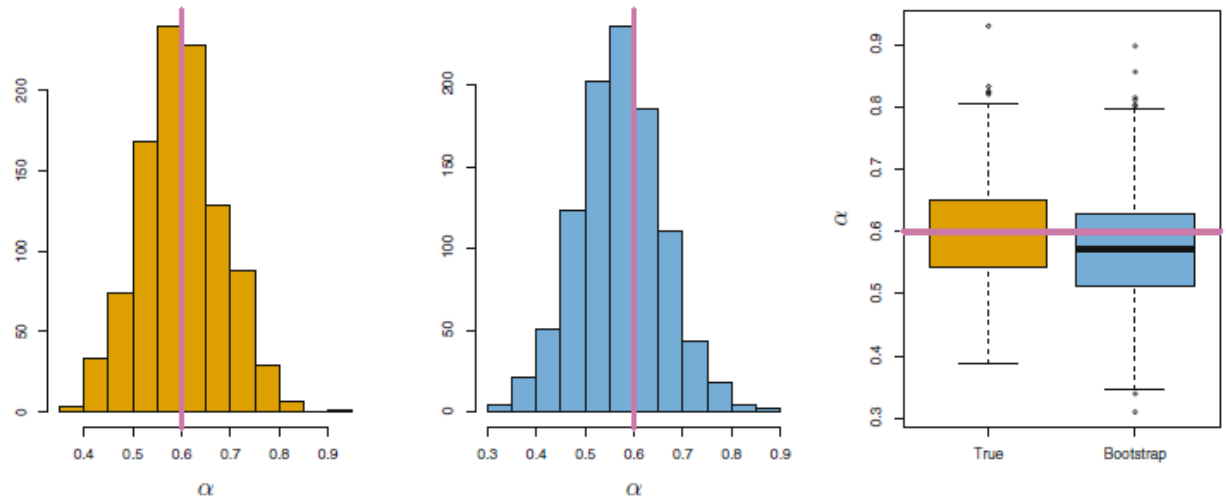


FIGURE 5.10. Left: *A histogram of the estimates of* α *obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of* α *obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of* α *displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of* α.

황색이 앞서 해본 모집단을 반복추출해서 얻은 결과이고, 청색이 하나의 data set만을 가지고 bootstrap으로 반복추출해서 얻은 결과이다. 각 sample에 대해 추정량 α를 구했을때, 이들의 분포가 매우 유사해보인다. 이는 오른쪽의 박스플롯을 통해서도 확인할 수 있다. 실제로 bootstrap을 통해 구해본 $SE(\hat{\alpha}) = 0.087$로, 모집단을 통해 구해본 0.083과 상당히 유사하다. 이러한 방식을 통해, 적은 수의 data set을 가지고 구하기 힘든 통계량의 특성까지도 구할 수 있게 되는 것이다.

참고

LOOCV의 variance가 크다 : https://stats.stackexchange.com/questions/178388/high-variance-of-leave-one-out-cross-validation

모델의 하이퍼파라미터를 test하는게 아니다 : https://stats.stackexchange.com/questions/11602/training-with-the-full-dataset-after-cross-validation?noredirect=1&lq=1

LOOCV와 k-fold의 bias variance(추천) : https://stats.stackexchange.com/questions/154830/10-fold-cross-validation-vs-leave-one-out-cross-validation?noredirect=1&lq=1