

1. Resampling-Validation(ISLR 175 쪽~184 쪽을 참고하여 작성하였습니다.)

참고 개념) 먼저 설명- 나중에 자세히 설명하는 세션 있습니다.

Lasso Regression(Least absolute shrinkage and selection Regression)

-L1 norm 을 제한하는 기법으로, L1 penalized regression 이라고도 불린다.

$$\hat{\beta}^{Lasso} = \min(\beta) \{ (y - X\beta)^T (y - X\beta) + \lambda \|\beta\| \}$$

-shrinks all regression coefficients ( $\beta$ ) toward zero & set some of them to zero

⇒ Parameter estimation 과 variable selection 의 효과를 한 번에 제공한다.

-> Lambda 만큼 목적식에 penalty 를 줌으로써 , Beta 의 범위를 규제하는 효과, overfitting 을 방지하는 역할도 한다.

-앞서 배우신 것처럼, Model 의 bias-variance 는 trade-off 관계입니다.

그리고 모든 모델에는 Parameter, Hyper-parameter 가 존재합니다.

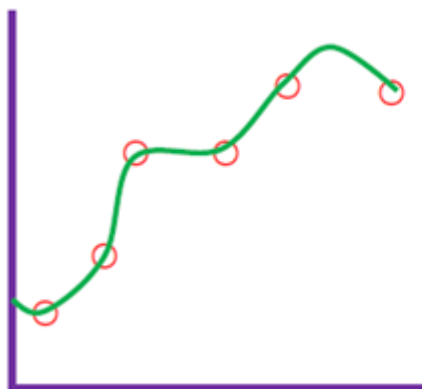
Lasso 를 예로 들면, Parameter 는 회귀분석의 Beta 와 같이 데이터로 인해 정해지는 것이고,

Hyper-parameter 는 Lasso 에서의 Lambda 와 같이 연구자가 결정하는 파라미터 입니다.

Lasso 에서의 Lambda 는 회귀선의 Smooth 한 정도를 정해주는 Hyper-parameter 라고 해서, Smoothing parameter 라고도 합니다.

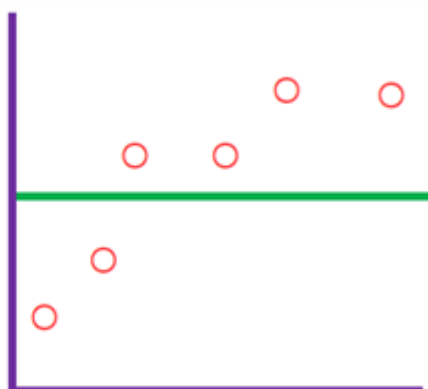
**Too Small lambda**

**-over-fitting 위험(high variance)**



**Too Large Lambda**

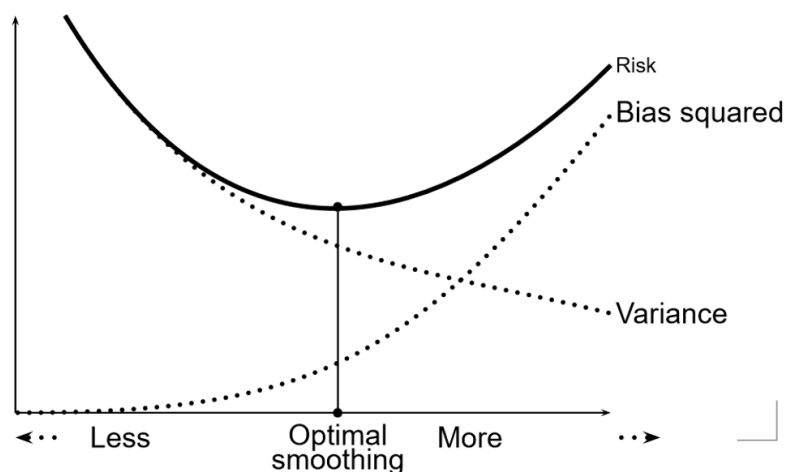
**-under-fitting 위험(high bias)**



Lasso regression 을 예시로 들어보자면, Lasso 는 Lasso 자체가 Linear regression 의 목적식에 L1-norm 을 추가하여, Beta 값들이 변할 수 있는 범위를 더 좁혀줘서, 모델의 variance 를 줄여주는 역할을 합니다, Lambda 가 클수록 더 크게 규제를 하기 때문에 너무 “Lambda 를 크게 주면” “오른쪽”과 같은 그림이 되고, “Lambda 를 너무 작게” 줘도 규제를 한 의미가 없이 “왼쪽”과 같은 회귀선이 됩니다. 그래서 이 Lambda 를 적절히 정해줘야 하는데 **이렇게 Lambda 를 적절히 정해줄 때 사용하는 방법이 바로 “Validation” 이라고 합니다.** Validation 은 아래 그림과 같이 Risk 또는 MSE 를 추정하는 개념입니다. 여기서의 MSE 는 New sample 에서의 error 값인데, 이는 알 수 없는 값이므로 “모수”와 같이 취급하고, 이를 추정하는 추정량으로써, 1) VALIDATION SET APPROACH 2) K-fold validation estimator 3) LOOCV estimator 등이 있습니다. 2번과 3번은 Training 과 Test 를 번갈아 바꿔가면서 Validation 을 한다 하여, 특별히 “Cross” validation 이라고 합니다.

그리고 각 Validation 방법을 통해, Optimal Smoothing, Lasso 에서는 최적의 Lambda 값을 결정할 수 있습니다.

## The Bias–Variance Trade off



$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Validation 을 하는 목적에는 크게 2가지가 있습니다.

- 1) 적절한 test set 이 없을때, test error 를 추정하는 것 (모수, 추정량의 개념)

2) 적절한 Smoothing parameter(level of flexibility)를 정하는 것

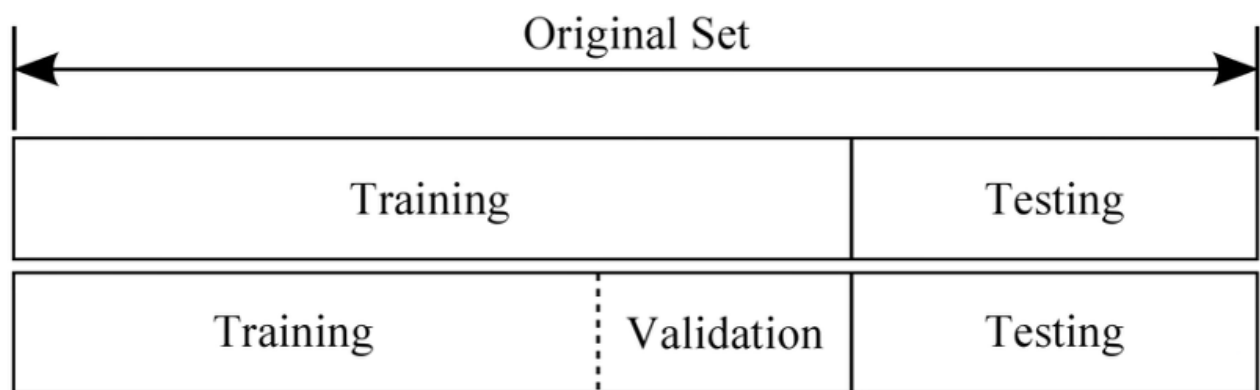
- 2번을 할 때 미리 떼어둔 Test set 을 이용한다면, Cheating(컨닝)을 하는것과 마찬가지이므로, 정말 new sample 에 대한 성능을 보장할 수 없게 됩니다.

다음은 3개 Validation 방법,

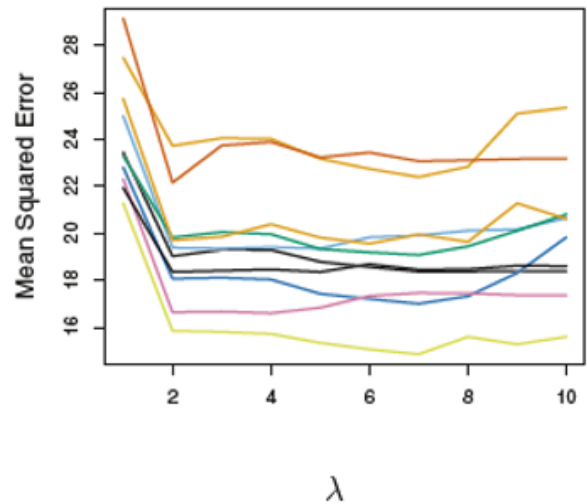
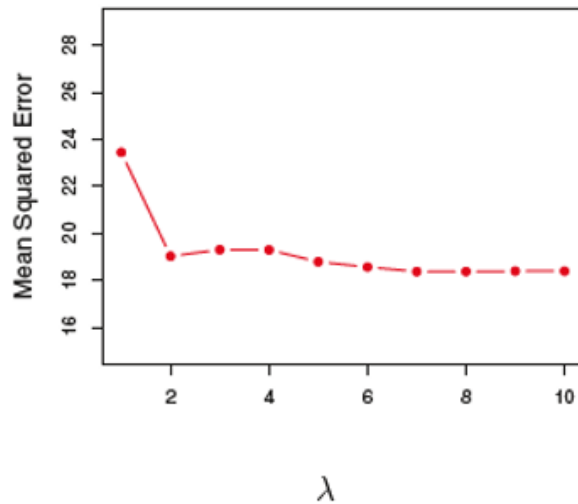
1) Validation set approach , 2) LOOCV(Leave-one-out CV), 3) k-fold cv

각 방법에 대한 구체적인 설명입니다.

#### 1) Validation set approach(단일 Validation set)



위의 그림 처럼 가지고 있는 데이터셋 중 랜덤하게 70% 정도를 train set 에 할당하고 나머지 30% 의 데이터를 test set 으로 나누고, "Training 셋에서 다시 30% 정도를 Validation set 으로 나누어서, 이 하나의 Validation set 을 이용하는 것이 Validation set approach 입니다."



1. 다음 그림 중 왼쪽을 보시면, Validation set 을 한 번만 나눴을때, **Lasso 에서 Lambda 를 증가시키기에 따른, MSE 의 추이**고, 오른쪽은 다양하게 나눴을때의 MSE 의 추이입니다. (그래프표시)- 여기서는 **Lambda=2가 최적점**입니다.

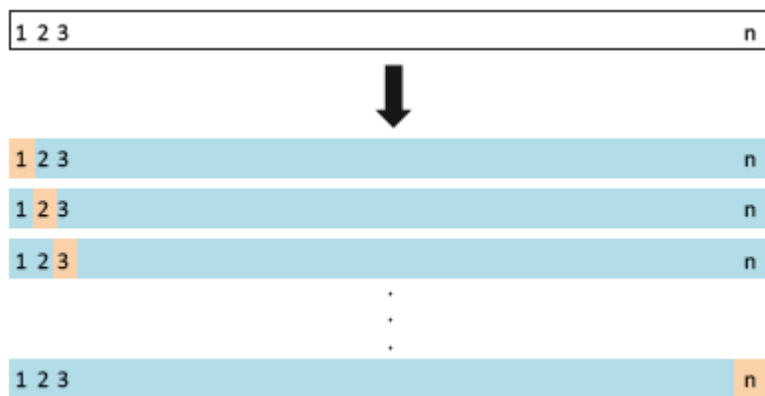
오른쪽을 보시면 어떤 관측치가 Training set/ Validation set 에 포함되는지에 따라서, 다양한 MSE 추정량이 나오는 것을 알 수 있습니다.

2. 1에서 알 수 있듯이, Validation approach 는 1번만 Validation 을 하기 때문에, **다른 MSE 추정 방법보다 확실히 덜 신뢰할 수 있고**, 무엇보다 Validation set 을 일부 떼어두어, **더 적은 관측치로 학습을 하므로, 여기서의 validation error 는 test error 를 Over-estimate 할 수 있다**는 단점이 있습니다.

## 2) LOOCV(Leave-One-OUT Cross-validation)

- 1) Validation set approach 의 단점을 보완하기 위해, LOOCV 방법이 나왔고,

이름 그대로, Training set 에서 1개의 관측치 씩 빼두고, 나머지 (n-1)개의 관측치로 학습을 한 뒤에 해당 관측치에 대한 estimation 을 하는 과정을 총 n 번 반복합니다. 한 번 추정할 때 해당 관측치에 대한  $MSE_1 = (y_1 - \hat{y}_1)^2$  를 계산하고, 이런 방식을 총 n 번 반복한 것을 더하고 데이터의 수 n 을 나누어, 최종 MSE 를 추정하게 됩니다.



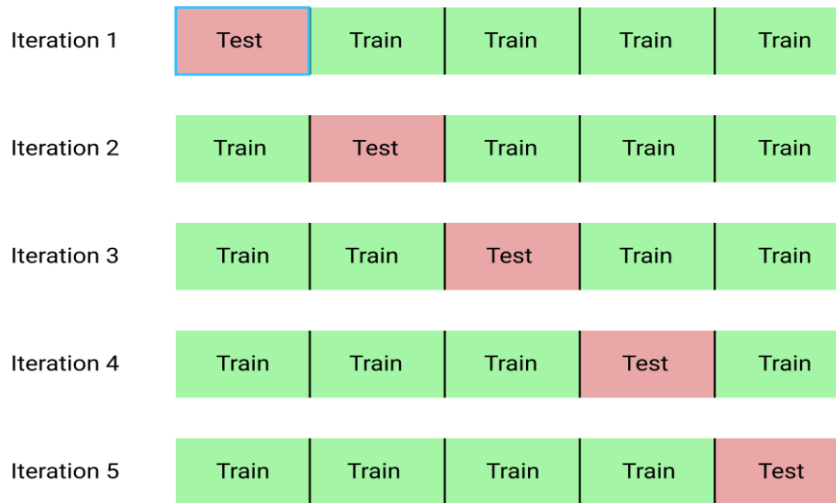
$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n \text{MSE}_i.$$

LOOCV의 Validation set approach에 대비한 장점은 다음과 같습니다.

첫째로) **훨씬 적은 bias**를 갖습니다. 여기서 말하는 “bias”는 모수인 “MSE 값”과 “추정된 MSE”와의 차이인데, Validation set approach는 앞에서 말씀드렸다시피, 일부(보통 Training의 30%)를 떼어놓고 학습을 하기 때문에, **Test error**를 과대추정하는 경향이 있습니다. 반면에, **LOOCV**는 **거우 1개 떼어놓고 학습을 하기 때문에, 데이터셋 전체를 사용해서 Test error를 구하는 것과 큰 차이가 없습니다.** 결론적으로, LOOCV Approach는 Validation approach와 달리, test error를 Over-estimate하지 않습니다.

둘째로) Validation approach의 단점으로, training, validation을 어떻게 나누냐 하는 랜덤성 때문에, 다르게 나눌때마다 Validation error가 다를 수 있고, 그래서 단일 Validation으로는 크게 믿을 수 없다고 말씀드렸습니다. 그러나 LOOCV는 위에서 이해하셨다시피, **Training, Validation을 나누는 개념이 아니라서, 항상 같은 Validation error를 냅니다.** 그래서 Validation approach보다 더 믿을만 합니다.

### 3) k-fold Cross-validation



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

-2) LOOCV 에 대한 대안이 k-fold CV 입니다. 이는 위와 같이, 데이터를 크기가 동일한 k 개의 그룹으로 나누고, 첫번째를 validation, 나머지 k-1개의 그룹을 training set 으로 활용합니다. 다음에는 2번째 그룹을 validation set 으로 쓰고 나머지를 training 으로, 이런 식으로 총 k 번 반복하게 되고 MSE 도 총 k 개 나옵니다. 최종적으로는, k 개의 MSE 를 k 로 나눠 평균 낸 값을 MSE 에 대한 추정값으로 사용합니다. LOOCV 는 k-fold 방법에서 K 가 데이터 개수만큼일때로 생각할 수 있습니다. 그렇다면 “k=n”인 LOOCV 에 대비한 장점은 다음과 같습니다.

1) LOOCV 는 n 번의 학습을 하기 때문에, 데이터 사이즈가 크다면, 컴퓨팅 시간이 오래 걸릴 수 있고, 10-fold cv 를 한다고 하면 n 번 대신 10번만 학습하면 되기 때문에 **Validation 에 드는 시간이 훨씬 절감됩니다.**(주의: LOOCV 도 MSE 는 1번만 추정됨, n 개의 MSE 가 나오는 것이 아니다. 관측치별로 예측치 1개임)

2) bias- variance trade-off for k-fold cv

LOOCV 는  $(n-1)$ 개의 학습 데이터를 사용하기 때문에, test error 에 대한 bias 는,  $\frac{n \cdot (k-1)}{k}$ 개를 이용하  
 는 K-fold cv 보다 훨씬 적습니다. 대신에, n 개의 모델을 이용하여 Validation 을 하기 때문에 high  
 variance 의 특징을 갖습니다. High variance 도 일반화라는 측면에서 좋지 않기 때문에, 그 중간에  
 있는 5-fold, 10-fold 정도를 사용하면 test error 에 대해 bias 나 variance 가 적절한 validation 을  
 할 수 있습니다.

#### 4) 총정리-3개 Validation 방법 총 정리 및 비교(그래프)

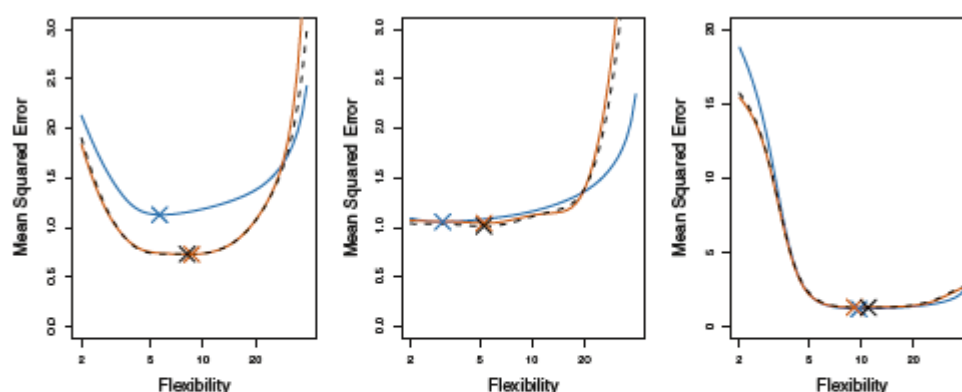
1) True test MSE : Blue line

2) LOOCV estimate for MSE : black dashed line

3) 10-fold CV for MSE: Orange line

\* x 표시: Validation 방법별 로, MSE 가 최소가 되게 하는 Smoothing parameter

- 설명: 파란색 라인이 test MSE 자체인 모수고, X 표시가 Smoothing parameter 의 최적점이라고  
 생각하면 됨.



**FIGURE 5.6.** True and estimated test MSE for the simulated data sets in Figures 2.9 (left), 2.10 (center), and 2.11 (right). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each of the MSE curves.

실제 데이터를 분석할때는, True test MSE, 즉, 정말 뉴 샘플이 왔을때의 MSE 를 알 수 없기 때문  
 에, 위에서 언급한 VALIDATION, K-FOLD, LOOCV 등의 추정에 대한 정확도를 알 수 없습니다.

하지만, 시뮬레이션 데이터에 한해서는 TEST MSE 를 정확하게 알 수 있는데 위의 그래프를 보시면, 파란색이 True test MSE, 검은색 라인이 LOOCV estimate, 오렌지색이 10-fold cv estimate 입니다. x 표시는 validation 방법별로 MSE 가 최소를 갖는 Smoothing parameter 값입니다.

각각이 다른 데이터인데, 세 플랏 모두 두 Validation의 추정값과 실제 test MSE가 매우 비슷합니다. 차이점은, 가운데는 low degree of flexibility(**large lambda**)에서는 test MSE를 매우 잘 추정하는데, high degree(**small lambda**)에서는 MSE를 과대 추정 하고 있습니다.

맨 왼쪽 플랏에서는, 모양 자체는 거의 잘 따라가는데, 전반적으로 true test MSE를 과소추정하는 것을 알 수 있습니다. 맨 앞에서 말씀드렸듯이, Validation의 목적은 크게 2개 입니다.

- 1) 마땅한 test set이 없을 때, test set을 추정하는 것 - estimated test error가 관심
- 2) 적당한 smoothing parameter를 정하는 것 - estimated test MSE가 최소가 되는 지점 (Smoothing parameter) 궁금할 수 있습니다. 1)에서는 정확하게 test error를 추정하는 것이 목적이지만, 2)에서는 정확한 test error값을 추정하기 보다는, 위에서의 그래프와 같이, MSE 곡선의 SHAPE을 잘 추정하여서, MINIMUM POINT가 어디인지가 주요 관심사가 됩니다. 그래서 왼쪽 플랏을 보시면, 전반적으로 TEST MSE를 과소추정하고 있지만, 경향은 같기 때문에, MINIMUM POINT는 비슷합니다. 여기서는 True test MSE를

최소로 만들어주는 smoothing parameter 지점이 관심사이므로, test error를 정확하게 추정하지 못했어도 잘 된 validation이라고 볼 수 있습니다.

\*But at other times we are interested only in the location of the minimum point in the estimated test MSE curve.

**\*사족: Regression 말고 Classification 문제에서의 VALIDATION 은 MSE 대신에 아래와 같이 오분류율을 이용하시면 됩니다.**

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n \text{Err}_i,$$

where  $\text{Err}_i = I(y_i \neq \hat{y}_i)$