4. Cross Validation
2019년 가을 학기

2019년 08월 28일
https://www.github.com/KU-BIG/KUBIG_2019_Autumn

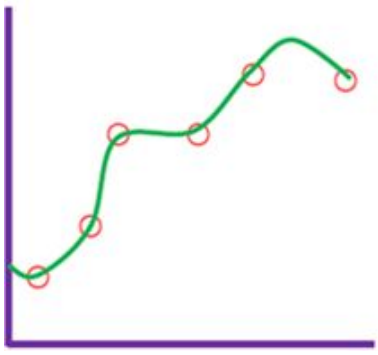
이 글은 ISLR p.175 ~ p.184를 참고하여 작성하였습니다.

1. Hyperparameter tuning을 위한 Validation의 필요성

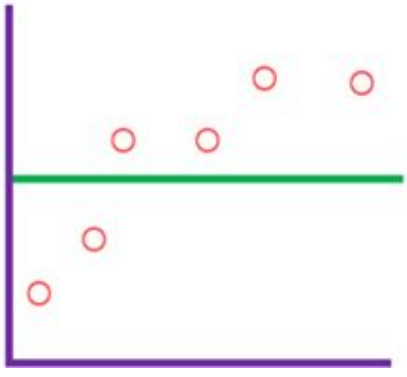
모델의 bias와 variance는 **trade-off** 관계이며, 모델에는 parameter, hyperparameter가 존재한다. Lasso regression(7강 regularized regression에서 자세히 다룬다)을 예로 들면, parameter는 회귀분석의 β(Lasso 회귀계수)와 같이 데이터로 인해 정해지고, hyperparameter는 Lasso에서 over-fitting을 방지하기 위한 λ와 같이 연구자가 결정하는 파라미터이다.

- Lasso에서의 λ는 회귀선의 Smooth한 정도를 정해주는 hyperparameter라고 해서, Smoothing parameter라고도 한다.

Too Small lambda
-over-fitting 위험(high variance)



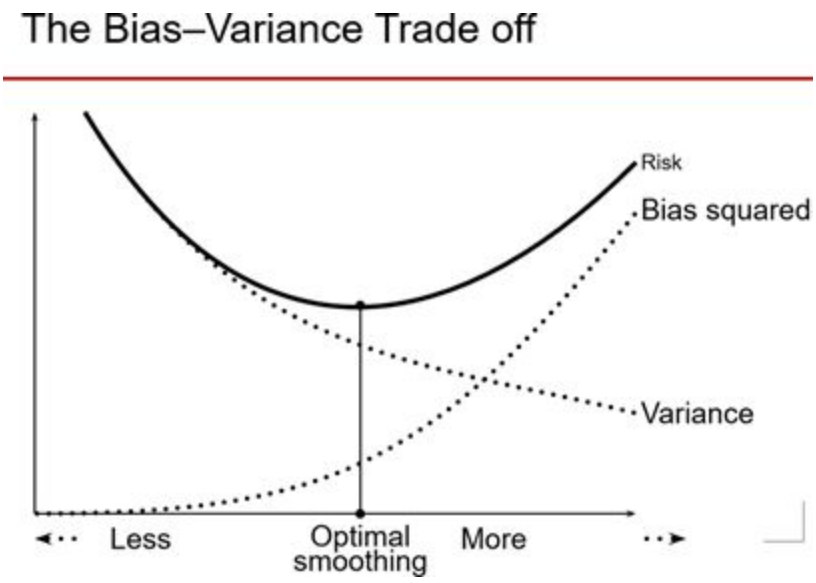
Too Large Lambda
-under-fitting 위험(high bias)



계속해서 Lasso를 예시로 들자면, Lasso는 선형회귀의 목적식(Sum of Square Error)에 규제(L1-norm)를 주어, β 값(회귀 계수)들이 변할 수 있는 범위를 제한해, 모델의 분산을 줄여주는 역할을 한다. λ가 너무 크면 규제 또한 커지기 때문에 오른쪽 그림과 같이 과소적합의 위험이 있고, λ가 너무 작으면 규제를 한 의미가 없어 왼쪽과 같은 과대적합의 위험성이 있다. 그래서 λ를 적절히 정해줘야 하는데 이때 사용하는 방법이 바로 validation이다.

Validation은 아래 그림과 같이 risk 혹은 MSE를 추정하는 개념이다. 여기서의 MSE는 새로운 표본에서의 error값인데, 이는 알 수 없는 값이므로 모수와 같이 취급한다. 이를 추정하는 추정법으로는 1. Validation set approach, 2. LOOCV estimator, 3. K-fold validation estimator 등이 있다. 2번과 3번은 training과 test를 번갈아 바꿔가면서 validation을 한다 하여, 특별히 Cross Validation(CV)이라 한다.

이러한 validation 방법을 통해 optimal smoothing값(Lasso의 최적의 λ값)을 결정할 수 있다.



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

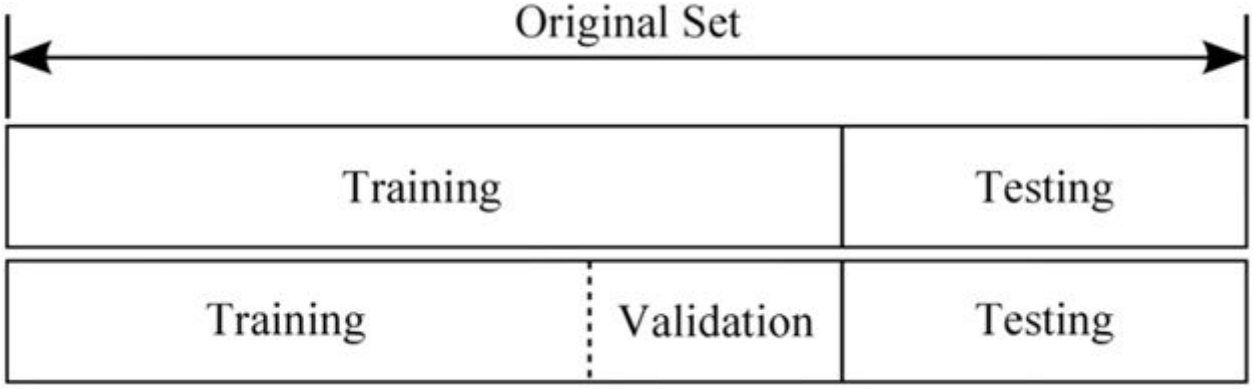
1.1 Validation 목적

1) 적절한 test set이 없을때, test error를 추정하는 것 (모수, 추정량의 개념)
   새로운 데이터가 주어졌을 때 모델이 얼마나 일반화 가능성을 가지고 작동하는지 확인한다.

2) 적절한 smoothing parameter(level of flexibility)를 정하는 것
   만약 K-fold CV로 smoothing parameter를 정할 때, 미리 떼어둔 test set을 이용한다면 cheating(컨닝)을 하는것과 마찬가지이므로 새로운 데이터에 적합 시 성능을 보장할 수 없다.

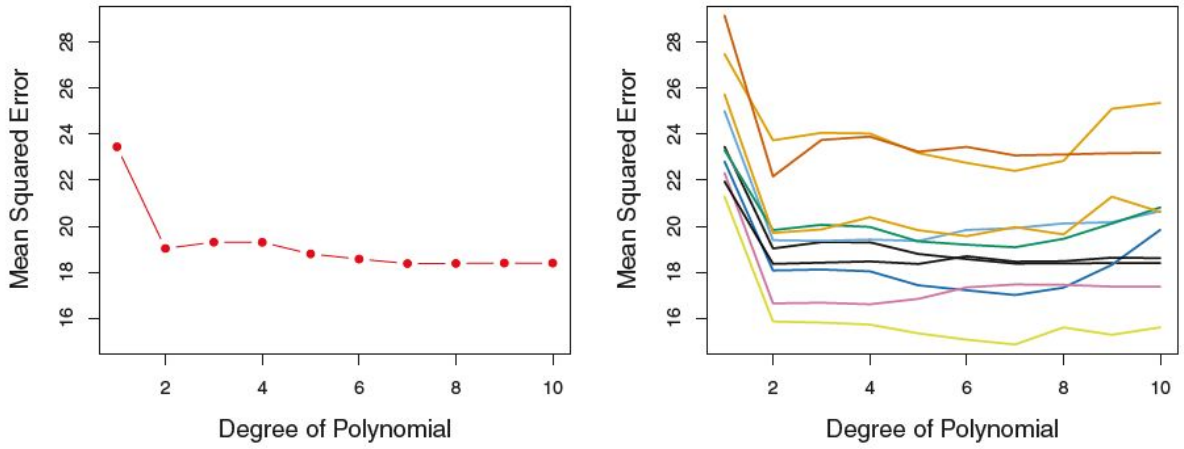
2. Validation 방법

2.1 Validation set approach (단일 Validation set)



Validation set approach의 과정은 다음과 같다. (아래 제시한 비율을 반드시 지킬 필요는 없다.)

1. 가지고 있는 데이터셋 중 랜덤하게 70% 정도를 training set에 할당한다.
2. 나머지 30%의 데이터를 testing set으로 나눈다.
3. Training set에서 다시 30% 정도를 validation set으로 나눈다.
4. 이 하나의 validation set을 이용하여 MSE를 추정한다.



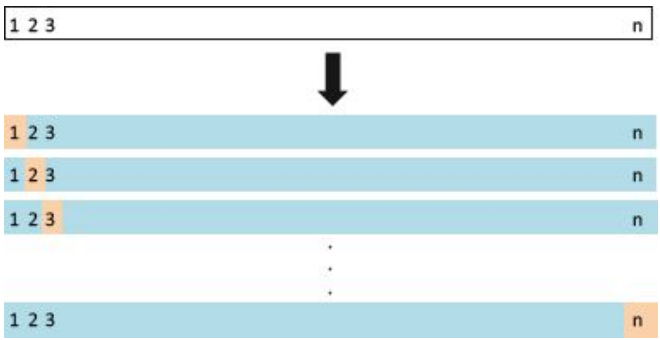
위 그림 중 왼쪽은 validation set을 한 번만 나눴을 때 polynomial regression에서 포함하는 설명변수의 최고차항(X, X^2, X^3....)을 증가시킴에 따른 MSE의 추이고 오른쪽은 validation set을 다양하게 나눴을때의 MSE의 추이다. 여기서는 포함하는 설명변수의 최고차항이 2일 때가 최적점이다. 오른쪽을 보면 어떤 관측치가 training set / validation set에 포함되는지에 따라서 다양한 MSE 추정량이 나오는 것을 알 수 있다.

Validation set approach의 단점은 validation을 한 번만 하기 때문에 다른 MSE 추정 방법보다 신뢰도가 떨어진다. 무엇보다 validation set을 일부 떼어두어 더 적은 관측치로 학습을 하므로, 여기서의 validation error는 test error를 과대추정(over-estimate)할 수 있다는 단점이 있다.

2.2 LOOCV(Leave-One-OUT Cross-validation)

Validation set approach의 단점(단일 측정으로 인한 낮은 신뢰도 / test error에 대한 과대추정 위험성)을 보완하기 위해, LOOCV 방법이 고안되었다. 그 과정은 다음과 같다.

1. Training set에서 1개의 관측치를 미리 빼고, 나머지 (n-1)개의 관측치로 학습을 한다.
2. 사전에 leave out된 관측치에 대한 estimation 과정을 총 n 번 반복한다.
3. 한 번 추정할 때마다 해당 관측치에 대한 $MSE_k = (y_k - \hat{y}_k)^2$ 를 계산한다.
4. 그렇게 계산된 n개의 MSE를 모두 더하고 총 데이터의 수 n으로 나누어, 최종 MSE 추정치를 계산한다.



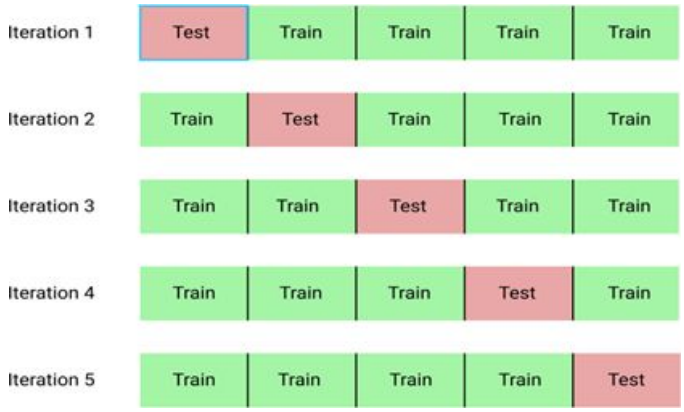
$$\mathrm{CV}_{(n)} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSE}_i.$$

LOOCV 방법의 Validation set approach에 대비되는 장점은 다음과 같다.

1. 훨씬 작은 bias를 갖는다. 여기서 말하는 bias란 모수인 MSE 값과 추정된 MSE 값의 차이다. Validation set approach은 일부(보통 training의 30%)를 떼어놓고 학습을 하기 때문에 test error를 과대추정하는 경향이 있다. 반면 LOOCV는 단 1개의 관측치를 떼어놓고 학습하기 때문에 데이터셋 전체를 사용해서 test error를 구하는 것과 큰 차이가 없다. 결론적으로 LOOCV는 validation set approach에 비해 test error를 과대추정할 위험성이 훨씬 줄어든다.

2. Validation set approach는 training set, validation set을 어떻게 나누는가에 따라 validation error가 달라진다. 단일 validation이 신뢰성이 낮은 이유이기도 하다. 이에 반해 LOOCV는 데이터를 training set, validation set으로 나누는 개념이 아니라서 항상 같은 validation error를 낸다. 따라서 LOOCV의 신뢰성이 validation approach보다 높다고 할 수 있다.

2.3 k-fold Cross-validation



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

LOOCV의 느린 속도를 보완한 대안이 k-fold CV이다. k-fold CV의 과정은 다음과 같다.

1. 데이터를 크기가 동일한 k개의 그룹으로 나누고, 첫번째를 validation set, 나머지 k-1개의 그룹을 training set으로 활용한다.
2. 다음으로 2번째 그룹을 validation set, 나머지를 training set으로 활용한다.
3. 이와 같은 과정을 총 k번 반복하여 k개의 MSE 추정값을 구한다.
4. 마지막으로 k개의 MSE를 모두 더한 뒤, k로 나눠 평균을 낸 값을 MSE에 대한 최종 추정값으로 사용한다. LOOCV는 k-fold 방법에서 k가 데이터 개수만큼인 경우라고 생각할 수 있다.

k-fold CV의 LOOCV에 대비되는 장점은 다음과 같다.

1. LOOCV는 n번의 학습을 하기 때문에 데이터 사이즈가 크다면 컴퓨팅 시간이 너무 오래 걸릴 수 있다. K-fold CV는 n번 대신 k번(k는 n보다 작거나 같은 수)만 학습하면 되기 때문에 validation에 드는 시간이 훨씬 절감된다.

   (주의: LOOCV도 MSE 추정치는 1개만 나옴. n개의 MSE 추정치가 나오는 것이 아니다.)

2. LOOCV는 (n-1)개의 학습 데이터를 사용하기 때문에, test error에 대한 bias는, $\frac{n}{k}(k-1)$개를 이용하는 k-fold CV보다 훨씬 작다. 대신 n개의 모델을 이용하여 validation을 하기 때문에 상대적으로 분산이 크다(high variance)는 특징을 갖는다. 높은 분산 또한 일반화라는 측면에서 좋지 않기 때문에, 그 중간에 있는 5-fold, 10-fold 정도를 사용하면 bias나 variance가 적절한 validation을 할 수 있다.

3. Validation방법 정리 및 비교

1) True test MSE : Blue line

2) **LOOCV estimate for MSE** : black dashed line

3) 10-fold CV for MSE: Orange line

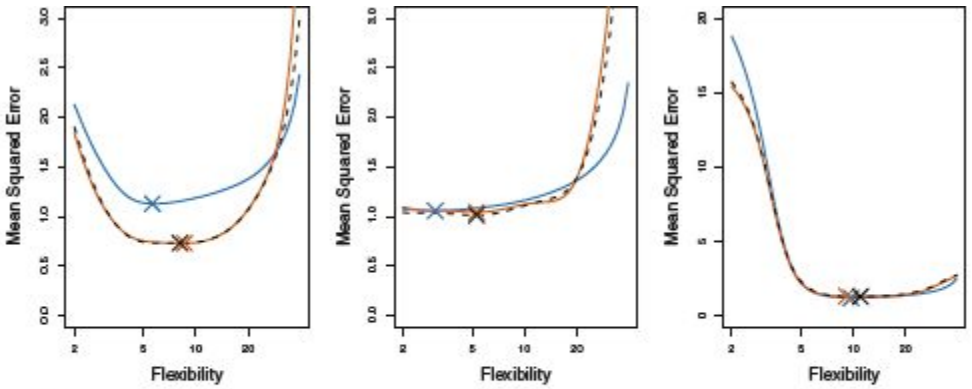


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9 (left), 2.10 (center), and 2.11 (right). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the 10-fold CV estimate is shown in orange. The crosses indicate the minimum of each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

실제 데이터를 분석할때는 true test MSE, 즉 진짜 새로운 샘플이 왔을때의 MSE를 알 수 없기 때문에, 위에서 언급한 validation set approach, LOOCV, k-fold CV 등의 추정에 대한 정확도를 알 수 없다. 하지만 시뮬레이션 데이터에 한해서는 test MSE를 정확하게 알 수 있다. 위의 그래프를 보면 파란색이 true test MSE, 검은색이 LOOCV estimate, 오렌지색이 10-fold cv estimate이다. X표시는 validation 방법별로 MSE가 최소가 되는 smoothing parameter 값이다.

세 그림 모두 각각 다른 데이터인데도 불구하고, 모두 두 개의 validation의 추정값과 실제 test MSE가 매우 비슷하다. 차이점이라 하면 가운데 그림의 low degree of flexibility(large lambda)에서는 test MSE를 매우 잘 추정된 것에 비해, high degree(small lambda)에서는 MSE를

과대 추정 하고 있다. 또한, 맨 왼쪽 그림에서는 모양 자체는 잘 따라가지만, 전반적으로 true test MSE를 과소추정하는 것을 알 수 있다.
앞서 언급한 바와 같이 Validation의 목적은 크게 2개 이다.

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

1)에서는 정확하게 test error를 추정하는 것이 목적이지만, 2)에서는 정확한 test error값을 추정하기 보다는, 위에서의 그래프와 같이, MSE 곡선의 모양을 잘 추정해, minimum point가 되는 지점이 어디인지가 주요 관심사가 된다. 왼쪽 그림을 보면, 전반적으로 test MSE를 과소추정하고 있지만 경향은 같기 때문에 minimum point가 되는 지점은 비슷하다. 여기서는 true test MSE를 최소로 만들어주는 smoothing parameter 지점(Lasso의 경우 λ값)이 관심사이므로, test MSE를 정확하게 추정하지 못했어도 성공적인 validation이라고 볼 수 있다.