Resampling

Bootstrap

Introduction

- 두 자산 X, Y에 투자를 한다고 할 때, risk를 가장 최소화하는 배분비율 α를 찾는 문제를 생각해보면, 아래와 같이 계산될 수 있다.

$$Var\left(\alpha X+(1-\alpha)Y\right)=\alpha^2 Var\left(X\right)+\left(1-\alpha\right)^2 Var\left(Y\right)+2\alpha\left(1-\alpha\right)Cov\left(X,Y\right)$$

$$\begin{aligned}\frac{\partial V}{\partial \alpha}&=2\alpha\sigma_X^2-2\left(1-\alpha\right)\sigma_Y^2+2\left(1-2\alpha\right)\sigma_{XY}\\&=2\alpha\left[\sigma_X^2+\sigma_Y^2-2\sigma_{XY}\right]-2\sigma_Y^2+2\sigma_{XY}\\&=0\\\therefore \alpha&=\frac{\sigma_Y^2-\sigma_{XY}}{\sigma_X^2+\sigma_Y^2-2\sigma_{XY}}\end{aligned}$$

- 하지만 population variance와 population covariance는 알 수 없으므로, 일반적으로 data를 통해 sample variance - sample covariance를 계산해 estimate하는 것이 일반적이다.
  - 그런데 이렇게 추정된 $\hat{\alpha}$ 역시 sample로부터 계산된 결과이므로, 이는 다시 말해 sample이 바뀌면 $\hat{\alpha}$값 역시 바뀔 수 있다는 것을 의미한다.
  - In other words, $\hat{\alpha}$는 sample로부터 계산되는 statistic이고, 그에 따라서 mean과 standard error를 가지며, 여러 번의 반복시행을 통해 true value로부터 추정치가 얼마나 떨어져 있는지를 파악할 수 있다.
  - 그리고 실제로 이렇게 **여러 개의 sample을 사용해 반복 추정**을 실시한 뒤, 모든 **estimate을 평균내면 true value에 아주 가까운 추정치**를 얻을 수 있게 된다.

Bootstrap

- BUT in practice 여러 개의 sample을 만들어낼 수 없다는 한계가 존재한다.
  - Original data만 사용하게 되면 딱 1개의 estimate $\hat{\alpha}$를 얻게 되므로, 이 estimate의 standard error, 즉 이것이 true value를 얼마나 잘 추정하는지 알 수 없다.
  - 이를 해결하는 방법이 bootstrap으로서, 하나의 sample로부터 같은 크기의 sample을 여러 개 만들어내는 방법이다.
- In other words, ***Bootstrap: 알아내고자 하는 parameter $\hat{\alpha}$의 표준오차를 구하는 방법***
- **Procedure of Bootstrap**

- 주어진 original data $\{x_i\}_{i=1}^n$가 있을 때, **sampling with replacement**를 통해 **크기가 n으로 똑같은 sample**을 만들어낸다. 이렇게 만들어낸 sample을 bootstrap sample이라 부른다.
  - 이렇게 만들어진 bootstrap sample은 ***original data와 not identical but follows the same distribution***
- 총 B개의 bootstrap sample을 각각 $S_1, S_2, \dots, S_B$라고 하면, 각각의 S_b로부터 추정하고자 하는 parameter $\hat{\alpha}_b$를 추정한다.
  - 단, 이때 **original data는 사용하지 않는다**.
- 이렇게 추정된 총 B개의 estimate $\hat{\alpha}_1, \dots, \hat{\alpha}_B$의 평균으로서 최종 estimate을 구하고, 그렇게 구한 **최종 estimate(즉 평균)과의 차이를 사용해 $\hat{\alpha}$의 표준오차를 대신한다.**

$$\bar{\hat{\alpha}} = \frac{1}{n}\sum_{b=1}^{B}\hat{\alpha}_b$$

$$SE_B\left(\hat{\alpha}\right) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\alpha}_b - \bar{\hat{\alpha}}\right)^2}$$

  - and replace $SE\left(\hat{\alpha}\right)$ with $SE_B\left(\hat{\alpha}\right)$.

- **Underlying assumption**
  - Original data의 모든 observation이 $i.i.d$을 따라야 한다.
    - Original data 역시 sample이고, 따라서 $i.i.d$를 따를 경우 original data의 분포가 True parameter의 분포와 같다.
    - Sampling-with-replacement는 $i.i.d$를 보장하는 sampling 방법으로, 따라서 이에 따라 sampling된 bootstrap sample 역시 original data의 분포, 나아가 True parameter의 분포를 따르기 때문이다.
  - $i.i.d$를 따르지 않는 데이터, 특히 time-series data의 경우 bootstrap을 사용할 수 없는 것은 아니지만, 단순히 sampling with replacement를 적용할 수는 없다.
- Bootstrap sample은 original data로부터 resampling하는 방법이기에 original data와 필연적으로 겹치는 부분이 발생한다.
  - BUT 평균적으로 개별 bootstrap sample에는 약 64%(or 2/3)에 해당하는 original observation만 포함된다.
  - pf) n개의 데이터로부터 n개를 뽑을 때, 1번의 trial에서 i번째 observation이 뽑힐 확률을 $\frac{1}{n}$이다.
    - 따라서 뽑히지 않을 확률은 $1 - \frac{1}{n}$이며, sampling with replacement를 따르므로 개별 observation을 뽑는 시행은 서로 독립이다.

        - 즉 n번의 시행을 반복했을 때 i번째 observation이 뽑히지 않을 확률은 $\left(1-\frac{1}{n}\right)^n$이다.
        - 여기에 극한을 취해주면 $\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n = \exp(-1) \approx 0.3679$
    - 따라서 평균적으로 약 36%의 데이터는 개별 bootstrap sample에 포함되지 않는다.

Validation

Training error vs Test error

- Training error: Fitting(training)에 사용한 데이터에 다시 model을 적용했을 때 계산되는 error
- Test error: Training에 사용하지 않은 데이터에 model을 적용했을 때 계산되는 error
    - 이때 어떠한 data point도 training에 사용되어서는 안 되며, training dataset의 어떠한 정보도 test set에 포함되어서는 안 된다.
- 일반적으로, model은 **"Training에서 발생하는 error를 최소화하는 방향으로" fit**되기 때문에, training error rate은 test error rate은 **dramatically underestimate**한다.
    - 특히 Training error를 극단적으로 줄이기 위해 model complexity를 높이게 되면 오히려 test error rate이 훨씬 높아지는(성능이 나빠지는) 현상이 발생하는데 이를 overfitting이라고 한다.
    - Overfitting을 해결하는 가장 좋은 방법은 test set의 크기를 늘리는 방법이지만, 현실적으로 쉽지 않다.
- **Validation**: Test set을 사용하지 않고도 training model의 성능을 파악할 수 있는 방법으로, training에 사용하지 않은 일부 데이터에 대해서 fitting한 결과를 적용함으로써 **test error를 추정**하는 방법

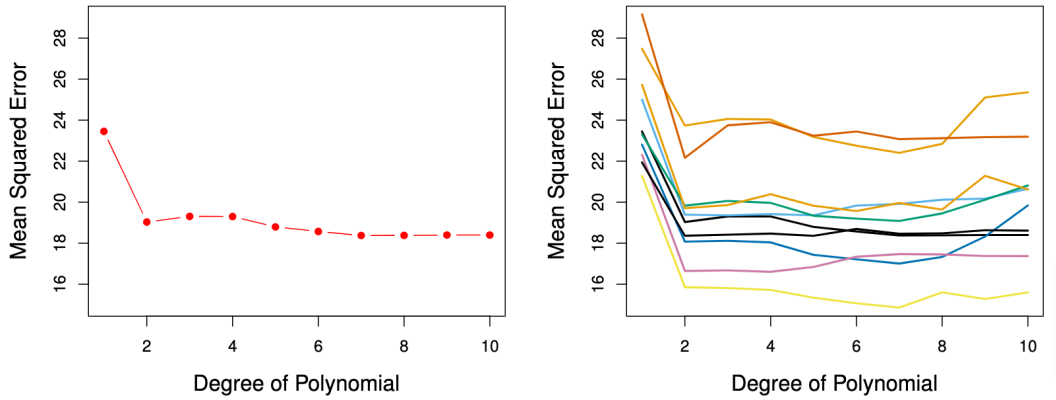
Hold-out validation

Procedure

1. 전체 주어진 sample(original data)를 일정 비율(typically 5:5 or 7:3 or 8:2)로 나누어 한 쪽을 training set, 다른 한 쪽을 validation set이라고 부른다.
2. Model fitting은 training set으로만 수행하며, 그렇게 fitted된 model을 validation set에 적용함으로써 성능을 파악하고 test error rate을 추정한다.
    - "정확하진 않더라도 이 정도의 성능을 보이겠다." or "이 정도면 test error 역시 낮을 것이다."

Drawbacks of hold-out approach

- Validation set의 결과 역시 "우연에 의한 것"일 수 있다.



  - 오른쪽의 결과처럼 validation set을 어떻게 구분하느냐에 따라서 test error에 대한 estimate이 robust하지 않을 수도 있다.
  - 그런데 왼쪽처럼 딱 하나의 validation set만을 사용할 경우, 해당 결과가 우연에 의한 것인지 혹은 training model의 true performance를 나타내는 것인지 알 수 없다.
- 특히 **주어진 데이터 전체를 training에 사용한 것이 아니므로**, model의 bias가 높을 수 있고 그에 따라서 **test error가 overestimate될 수 있다.**

Cross-validation

- Hold-out의 장점(test data를 사용하지 않고도 model 성능을 파악할 수 있다)을 살리면서 단점(주어진 data 전체를 training에 사용하지 않는다)을 줄이기 위한 방법
- 미리 정의된 K개만큼 전체 데이터를 등분한 뒤, K번 반복해서 데이터를 fitting함으로써 test error를 추정하는 방법

Procedure

1. 주어진 전체 sample을 미리 정의된 K개로 등분한다.
   - 이때 등분된 데이터들 사이에는 ***서로 겹치는 부분이 있어선 안 된다***.
2. 사용할 model $\hat{f}$을 설정한다.
3. 이렇게 등분된 subdata를 $E_1, E_2, \ldots, E_K$라고 하면, E_1부터 E_K까지 K번 반복을 돌면서
   - E_k를 validation set으로 사용하고, 나머지 $K-1$개 데이터 $E_1, \ldots, E_{k-1}, E_{k+1}, \ldots, E_K$의 묶음을 training set으로 사용한다.
   - Training set을 사용해 $\hat{f}$를 fitting하며, fitted model을 각 단계에서의 validation set E_k에 적용해 error를 계산한다.
4. 이렇게 계산된 K개의 error의 전체 평균으로서 최종 test error rate을 추정한다.

$$CV_{(K)} = \sum_{k=1}^{K} \frac{n_k}{n} MSE_k$$
$$\text{where } MSE_k = \frac{1}{n_k} \sum_{i \in E_k} (y_i - \hat{y}_i)^2$$

- MSE_k를 바로 plug-in하면 $CV_{(K)}$는 결국 subdata에서 계산된 RSS를 모든 subdata에 걸쳐서 더한 뒤, 전체 original observation으로 나누는 값으로서, 결국 이론적으로는 training error와 같은 결과를 낳는다.
- 단, 이때 $n_k = \frac{n}{K}$라면, $CV_{(K)}$는 $\frac{1}{K} \sum_{k=1}^{K} MSE_k$와 같아진다.

LOOCV

- LOOCV란 $n_k = 1$ 또는 $K = n$인 cross-validation이라고 볼 수 있다. 즉, 딱 1개의 data point만을 validation에 사용하고 나머지 $n - 1$개를 training에 사용하는 방법을 n번 반복하는 셈인 것이다.
- 그러나 current model t에서 사용하는 training data가 previous model $t - 1$에서 사용한 training data와 딱 2개(current validation point & previous validation point)를 제외하면 모두 동일하며,
  - 그렇기 때문에 각각의 fold에서 계산된 결과의 상관관계가 높고, 그에 따라서 model variance가 높다는 한계가 있음
- 또한 LOOCV의 경우 shuffle을 하는 것의 의미가 거의 없다.

Other issues

- CV 역시 전체 데이터를 training에 사용하는 것이 아니기 때문에, error가 overestimate될 가능성이 있다.
  - But hold-out에 비해서 그 가능성이 낮으며, 결론적으로 보면 CV를 수행한 이후 training에 사용된 적 없는 data point는 존재하지 않게 된다.
- LOOCV의 경우 model variance가 높은 만큼 bias가 낮으며, typically $K = 5$ or $K = 10$이 선호된다.
- 한편, test error estimate의 standard error를 구할 때 MSE_k와 $CV_{(K)}$를 사용하는 것이 useful하지만, valid하지는 않다.
  - CV에서는 validation set으로 사용되는 data를 제외하면 대부분의 training data를 그대로 돌려쓰기 때문에 error들 사이의 상관관계가 높고, 따라서 $i.i.d$ 가정에 위배되기 때문이다.
  - In other words, 각각의 MSE_k가 truly CV로부터 떨어진 것인지 다른 training set과의 관계 때문인 것인지 decompose하기 어려움
- Variable selection된 데이터에 대해서 Cross validation을 수행할 수 없다!

- Why? 가장 관련이 높은 predictor를 선택하는 과정에서 이미 training data를 사용했고, 이 과정 역시 training의 일부분이기 때문!
- 따라서 Variable selection을 먼저 진행한 뒤, 그렇게 select된 predictor만을 가지고 cross-validation을 하게 되면
  - Selection을 할 때 사용된 정보를 다시 training과 validation에 사용하기 때문에 overfitting이 발생할 가능성이 아주 높다.
- Variable selection 역시 training의 일부분이기 때문에, selection & training의 모든 단계에 걸쳐서 cross-validation을 진행해야 한다.
  - 즉, K-fold로 먼저 나눈 다음, $K-1$개의 training set에 대해서 variable selection & training을 수행한 뒤, 다시 validation set에 대해서 performance를 측정한다.
    - 개별 fold에서 선택된 variable & model을 validation set에 그대로 적용해주는 것

Bootstrap VS Cross-validation

- Test error estimation에 bootstrap을 사용할 수 있는가? NO!
  - Cross-validation에서 가장 중요한 부분은 Training - Validation set 사이에 **overlap이 없어야 한다는 것**!
  - Why not? Validation을 통해 performance를 측정할 때 training에서 사용된 정보가 반영되면 test error를 underestimate할 수 있기 때문!
  - 만일 bootstrap으로 error estimation을 한다면, bootstrap sample로 training하고, original data를 validation set으로 사용하는 방법을 생각할 수 있다.
    - BUT **bootstrap은 필연적으로 original data와 겹치는 부분이 발생**할 수밖에 없으며, 따라서 이는 validation set에 training data의 정보가 반영된다는 것을 의미한다.
      - 따라서 bootstrap sample로 test error를 추정하면 상당한 underestimation이 발생!
    - 특히, Bootstrap은 한정된 정보로 “쥐어 짜내는" 개념이기 때문에, technically 정보의 양이 더 늘어나지는 않으며, 따라서 estimator의 추정치가 더 정확해지는 것도 아니고, training 성능이 더 좋아지지도 않는다!
    - 단지 Bootstrap을 통해서는 **estimator의 variability를 알 수 있을 뿐**임
- **Test error 추정에는 cross-validation, 그렇게 추정된 error의 uncertainty 추정에는 bootstrap을 사용하는 것이 낫다.**