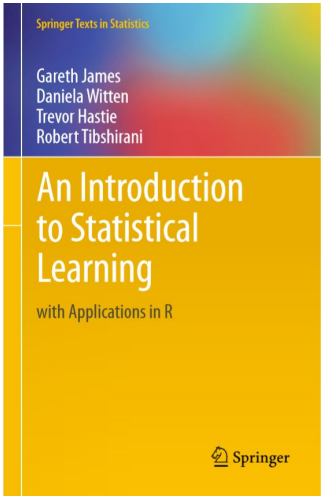
쉽게 배우는 머신러닝

Ch.5 RESAMPLING METHODS

훈 러닝 (Hun Learning)

January 7, 2020

TEXTBOOK



- **An Introduction to Statistical Learning : with Applications in R**
- 목차:
  1. Intro
  2. Statistical Learning
  3. Linear Regression
  4. Classification
  5. Resampling Methods
  6. Linear Model Selection and Regularization
  7. Moving Beyond Linearity
  8. Tree-based Methods
  9. Support Vector Machines
  10. Unsupervised Learning

INTRO

Resampling?

- **주어진 데이터서 반복적으로 "Resample",** 왜?
  ex) 주어진 데이터에 일단 fitting를 하긴 했는데, 이게 다른 데이터를 집어넣으면 얼마나 널뛰기할까가 궁금하다.
  → 여러 데이터로 fitting을 해 회귀계수를 마니마니 구해보자. 이것들을 보면 회귀계수의 sampling variability에 대해 가늠할 수 있다. 근데 난 데이터 하나밖에 없는데?
  → **데이터를 쪼개자!!**
- 하나의 데이터를 여러 개로 쪼개 fitting을 여러 번 한다(추정치를 여러 번 구한다.).
  모델의 성능과 추정치의 분산에 대해 더 많은 것을 알 수 있다!

대표적으로 쓰이는 방법은 **Cross-Validation, Bootstrap**

1. **Cross-Validation** : 데이터를 train/test 셋으로 여러 번 나눠 여러 번 test MSE를 구해, 1) 이 모델이 잘 맞는가 2) 어느 정도로 빡세게 fitting해야 하는가를 알아보자.
2. **Bootstrap** : 데이터에서 새로 랜덤으로 추출한 미니 데이터로 추정치를 잔뜩 구해, 데이터에 따라 추정치가 얼마나 널뛰는가 보자.

CROSS-VALIDATION

Test MSE가 낮은 모델이 "좋은" 것. 그러나 우리가 아는 것은 Train MSE이며, 이 둘은 엄연히 다르다. Test MSE를 추정할 방법은?

1. The Validation Set Approach

- 데이터를 Training set / Validation set으로 반반 나눠보자. Validation MSE가 Test MSE의 추정치이다. fitting 방법을 달리하면서 이 Validation MSE가 어떻게 변하는지 보자!

- 이처럼 Training set / Validation set을 나누는 과정을 여러 번 반복하면 여러 개의 Test MSE 추정치를 얻을 수 있다.

1. The Validation Set Approach

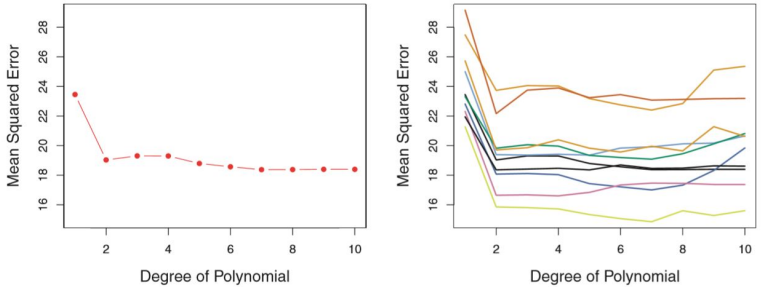


FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *Validation error estimates for a single split into training and validation data sets.* Right: *The validation method was repeated ten times, each time using a different random split of the observations into a training set and a validation set. This illustrates the variability in the estimated test MSE that results from this approach.*

1. The Validation Set Approach

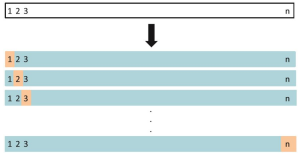
그러나 Validation Set Approach에는 다음과 같은 단점이 있다.

1. 내가 데이터를 어떻게 나누냐에 따라 Validation MSE가 크게 달라진다.
2. 관측치의 일부분, 예컨대 절반만 train set에 들어간다는 것이 치명적인 단점. 데이터가 수가 적을수록 모델 fitting이 잘 안 맞기 때문에(Bias) 이걸로 구한 Validation MSE가 실제 Test MSE보다 더 높게 나올 수 있다(Overestimate).

위 문제점을 보완한 것이 LOOCV(Leave-One-Out Cross-Validation)

2. LOOCV(Leave-One-Out Cross-Validation)

- 쉽게 말하면 Validation Set이 관측치 달랑 하나고, 거기서 $MSE_i = (y_i - \hat{y}_i)^2$를 구한다.



- 그걸 n번 반복해서 구한 것을 Test MSE의 추정치로 삼는다! 이렇게 하면 좋은 점은...

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

1. **Training / Validation set 구분에 따라 추정치가 달라질 일이 없다. 한번씩 다 쪼개니까!**
2. **n-1개로 fitting하니 n개를 사용하는 것과 관측 수에 따른 영향이 사실상 같다. 때문에 상대적으로 Test MSE를 과대추정하지 않는다!**

2. LOOCV(Leave-One-Out Cross-Validation)

- LOOCV는 다 좋은데 모델 fitting을 n번 해야하니 연산이 빡세다. 그러나 OLS 회귀분석에서는 아래 식으로 퉁치면 된다. 왜 그럴까? ($h_{ii} = [H]_{ii}$)

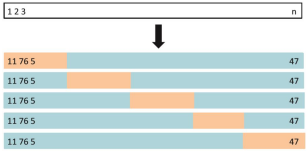
$$CV_{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right) \quad \left(\because y_i - y_{i(i)} = \frac{y_i - \hat{y}_i}{1 - \hat{h}_{ii}}\right)$$

- Y를 원래의 데이터, $\tilde{Y}$를 y_i를 $\hat{y}_{i(i)}$로 대체한 데이터로 생각해보자. 이때 $\hat{y}_{i(i)} = X_i^T \hat{\beta}_{(i)} = [H\tilde{Y}]_i$ 임에 주목하자*(why?)*. 때문에 다음과 같은 관계가 성립한다.

$$\begin{aligned}\hat{y}_i - \hat{y}_{i(i)} &= [HY]_i - [H\tilde{Y}]_i = [H(Y - \tilde{Y})]_i \\ &= \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & \cdots & h_{1,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{i,1} & h_{i,2} & h_{i,3} & \cdots & \cdots & h_{i,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{n,1} & h_{n,2} & h_{n,3} & \cdots & \cdots & h_{n,n} \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ y_i - \hat{y}_{i(i)} \\ \vdots \\ 0 \end{pmatrix} = h_{ii}(y_i - \hat{y}_{i(i)})\end{aligned}$$

3. k-Fold Cross-Validation

- 그러나 회귀분석 외에 한 번 fitting이 빡센 모델의 경우, 예컨대 관측치가 몇 만개가 넘어갈 때 어느 세월에 LOOCV를 구하고 앉았나. → **k개의 뭉텅이(batch)로 나누자!** (LOOCV는 k=n인 경우)



- 이렇게 해서 한 batch마다 MSE를 다 구하고 아래 식을 구해 Test MSE의 추정치로!

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

- 이렇게 하면 연산 부담이 덜하다는 장점도 있으나, 그보다 Test MSE 추정치의 분산이 더 줄어든다는 것이 특히 중요! 왜? **Bias-Var Tradeoff for k-Fold CV**

3. k-Fold Cross-Validation

- **Bias:** Training set에 관측치가 많을수록 모델의 Bias가 줄어든다.
- **Variance:** LOOCV의 경우 사실상 Validation MSE_i가 거의 동일한 Training set에서 구한 모델 추정치를 사용한다. 때문에 fitting도 거의 비슷하므로 MSE끼리 상관관계가 높을 것. 이러면 Test MSE의 추정치가 가지는 분산이 커질 수 밖에 없다.
  - 직관적으로 이해하자면, 나름 각기 다른 training set으로 n번 fitting하긴 하는데 그 차이가 미미하니 전체적으로 보면 fitting 결과가 그냥 똑같아져 버리는 것. 때문에 실질적으로 하나의 데이터 셋에서 fitting한 거와 마찬가지니까, 당연히 다른 데이터로 LOOCV를 구하면 더 값이 많이 달라진다.

$$\begin{aligned} Var(CV_{(k)}) &= Var(\frac{1}{k}MSE_1 + \frac{1}{k}MSE_2 + ... + \frac{1}{k}MSE_k) \\ &= \frac{1}{k^2}[Var(MSE_1) + Var(MSE_2) + ... + \sum_{i \neq j} Cov(MSE_i, MSE_j)] \end{aligned}$$

- k를 늘린다는 것은 더 잘게 쪼갠다는 것. 때문에
  Training set 안에 더 많은 관측치가 들어가므로 Bias ↓, Training set끼리 점점 유사해져 Variance ↑
- 통상 k=5, 10 정도를 쓰면 적절하다고 합니다.

3. k-Fold Cross-Validation

k=10으로 해도 대충 비슷하다!

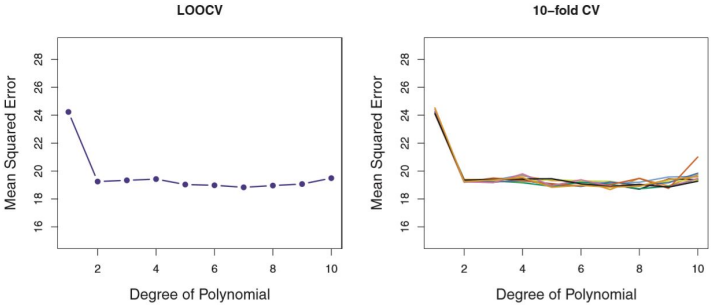


FIGURE 5.4. *Cross-validation was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`. Left: *The LOOCV error curve.* Right: 10-*fold CV was run nine separate times, each with a different random split of the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

3. k-Fold Cross-Validation

레벨은 틀려도 Test MSE가 최소화되는 지점은 얼추 잘 추정한다!

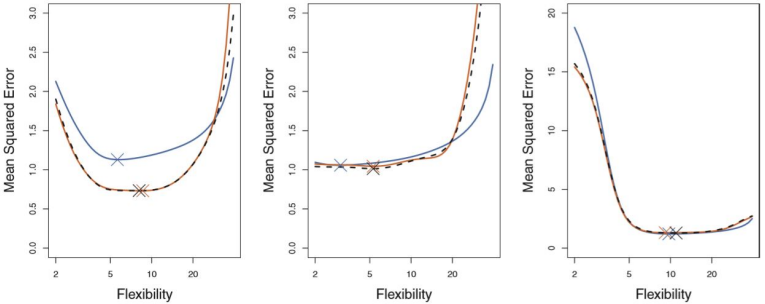


FIGURE 5.6. *True and estimated test MSE for the simulated data sets in Figures 2.9 (*left*), 2.10 (*center*), and 2.11 (*right*). The true test MSE is shown in blue, the LOOCV estimate is shown as a black dashed line, and the* 10*-fold CV estimate is shown in orange. The crosses indicate the minimum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

CROSS-VALIDATION

Cross-Validation: Classification

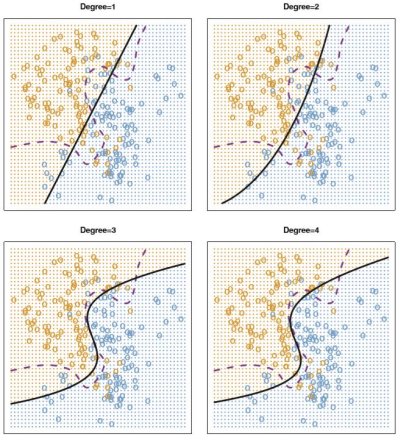
MSE 말고 Error Rate를 추정한다는 거 말곤 똑같음.

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

Example: Logistic Regression (Binary, p=2)

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2$$

BDB(보라색 점선)에 맞추기 위해서 고차항을 추가.
실제로는 BDB가 어디있는지 모르는데 어떡하나?
k-fold CV로 Test ER를 *추정하자*!



Cross-Validation: Classification

레벨은 틀려도 Test ER이 최소화되는 지점은 얼추 잘 추정한다!

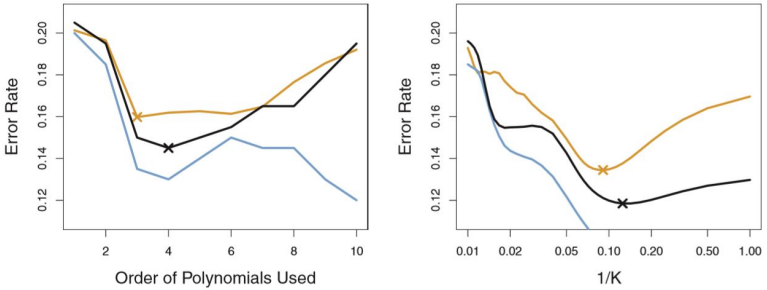


FIGURE 5.8. *Test error (brown), training error (blue), and* 10*-fold CV error (black) on the two-dimensional classification data displayed in Figure 5.7.* Left: *Logistic regression using polynomial functions of the predictors. The order of the polynomials used is displayed on the x-axis.* Right: *The KNN classifier with different values of K, the number of neighbors used in the KNN classifier.*

THE BOOTSTRAP

Bootstrapping?

1 Get (oneself or something) into or out of a situation using existing resources.
'the company is bootstrapping itself out of a marred financial past'

Figure: Bootstrap의 의미 (OXFORD)

- **통계학에서 Bootstrapping은 "Random sampling with replacement(복원추출)"**
- 모분포(True Population)의 모수 θ에 대한 추정통계량인 표본 X의 함수 $\hat{\theta}(X)$의 분포를 알고 싶다. 그러나 $\hat{\theta}(X)$의 분포에 대해 진짜 아무것도 모를때, 마치 우리가 모르는 $\hat{\theta}(X)$의 분포에서 생성된 것과 같은 sample $\hat{\theta}_1, \hat{\theta}_2, \hat{\theta}_3, ...$을 추가로 생성할 수 있다면?
- **이 때 만일 개별 표본이 iid 가정을 만족한다면, 표본의 Empirical Distribution을 모분포의 근사분포로 생각할 수 있음.** 이에 착안하여 이미 추출된 Sample에서 다시 반복 추출을 하고, 새로 추출된 $X_{resampled}$로 $\hat{\theta}(X_{resampled})$를 계산하는게 Bootstrapping!

Example: Optimal Asset Allocation

- 주어진 포트폴리오 수익률 수준에서 risk(분산)을 최소화하는 최적 비중 w를 찾아보자.

$$\overline{R_p} = wR_X + (1-w)R_Y$$

$$Min\ Var(\overline{R_p}) = w^2\sigma_X^2 + (1-w)^2\sigma_Y^2 + 2w(1-w)\sigma_{XY}^2$$

- 위 식의 우변을 w에 대해 미분하여 전개하면 다음의 최적 비중을 얻는다.

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

- 이때 n개의 과거 수익률 관측치 (R_{Xi}, R_{Yi})로 w의 추정치 $\hat{w}(R_X, R_Y)$를 얻을 수 있다.
- 그렇다면 모수 w의 추정량인 $\hat{w}$의 분포는?
  즉 우리의 추정치 $\hat{w}$의 sampling variability는 어떻게 가늠할 수 있을까?

Example: Optimal Asset Allocation

모수를 죄다 알고 있다면? 그냥 이런거 1,000개 뽑아서 $\hat{w}$ 1,000개 계산하면 되지.

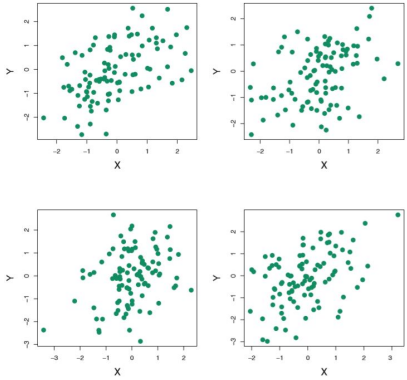
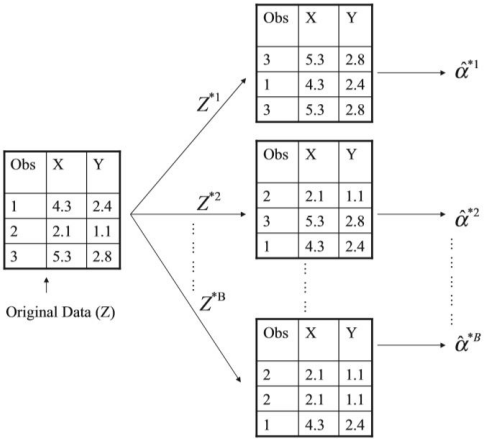


FIGURE 5.9. *Each panel displays* 100 *simulated returns for investments* X *and* Y. *From left to right and top to bottom, the resulting estimates for* α *are* 0.576, 0.532, 0.657, *and* 0.651.

Example: Optimal Asset Allocation

하지만 모르니까 아래 그림처럼 하나의 데이터셋 (R_X, R_Y)에서 1,000개의 bootstrap sample을 얻는다!



Example: Optimal Asset Allocation

놀랍게도 Bootstrap 한 결과와 모분포에서 sampling한 결과가 많이 비슷하다!

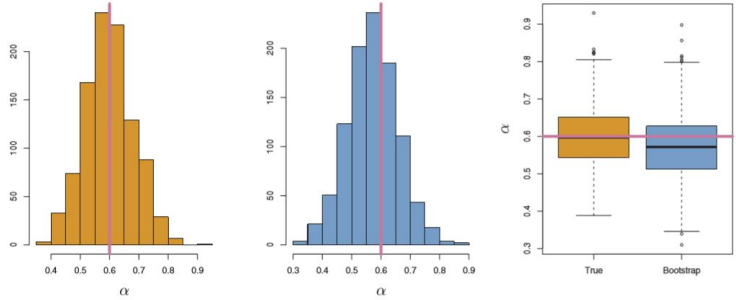


FIGURE 5.10. Left: *A histogram of the estimates of* α *obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of* α *obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of* α *displayed in the left and center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