Resampling methods

하나의 데이터에서 반복적으로 sample을 만들어 model을 적합시켜 보는 것. 하나의 training set에 모델을 적합 시킬 때보다 더 많은 정보를 얻을 수 있다. 여러개의 샘플을 사용해 회귀를 적합해보면 추정치가 어느정도의 범위에 있을지 알 수 있다. 근데 일반적으로 우리가 가지고 있는 샘플은 하나.. 그래서 나온 것이 resampling

[대표적인 방법 두가지]

1. **cross-validation**

   - 데이터를 train/validation으로 쪼개서 모델을 여러개의 validation MSE를 구한다
   - test set을 구할 수 없는 상황에서 test MSE를 추정하기 위해 validation set을 사용하는 것!

1. **bootstrap**

   - 우리가 가지고 있는 data에서 복원추출로 샘플을 계속 뽑아 추론하는 방법

1. cross-validation

- 모델의 최종목적 test error를 줄이는 것(내가 아는 데이터로 모델을 만들어서 모르는 데이터가 들어왔을 때 얼마나 잘 예측하는가). 하지만 test data를 가지고 있지 않은 경우가 다수 ex 내일 주식 예측, 이번연도 말에 있는 KBO 경기의 승률을 예측하는 경우
- 우리가 이미 가지고 있는 데이터를 나눠서 그 일부를 test data처럼 사용. validation set을 통해 validation MSE를 구하고 이를 test MSE의 추정치로 사용하자
- 우리가 가지고 있는 데이터를 최대한 활용하자!!

1.1 the validation set approach : 데이터를 반반 쪼개자

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.*

- 데이터를 random하게 반으로 잘라서 train set validation set으로 나누는 방법으로 이 과정을 반복하면 여러개의 Validation MSE를 구할 수 있다

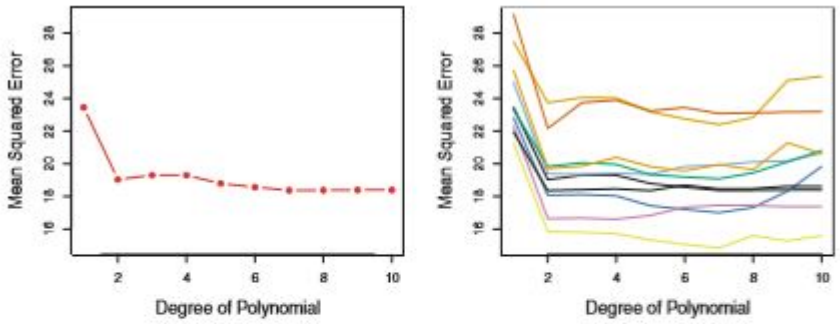

FIGURE 5.2. *The validation set approach was used on the* `Auto` *data set in order to estimate the test error that results from predicting* `mpg` *using polynomial functions of* `horsepower`*. Left: Validation error estimates for a single split into training and validation data sets. Right: The validation 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.*

- 오른쪽의 그림은 데이터를 나누는 것을 10번 반복해서 얻는 validation set MSE. 10개의 MSE 모두 1차항에서 2차항으로 넘어갈 때 MSE가 확연하게 줄고, 그 이후의 고차항을 사용하는 것이 큰 이득이 없음을 확인할 수 있다
- 여기서 나타나는 문제점 두가지:

1. 어떤 데이터가 validation set에 포함되어 있는지에 따라서 추정된 test MSE의 변동이 너무 크다 > 추정치로 사용하기에는 무리가 있음
2. 가지고 있는 데이터의 반만 사용해 fitting을 하기 때문에 validation MSE가 큰 값을 가질 수 있다(Bias가 커진다). 즉 Test MSE를 overestimate하게 된다.

1.2 Leave - One - Out Cross Validation (LOOCV) : 데이터를 하나 남기고 사용하자

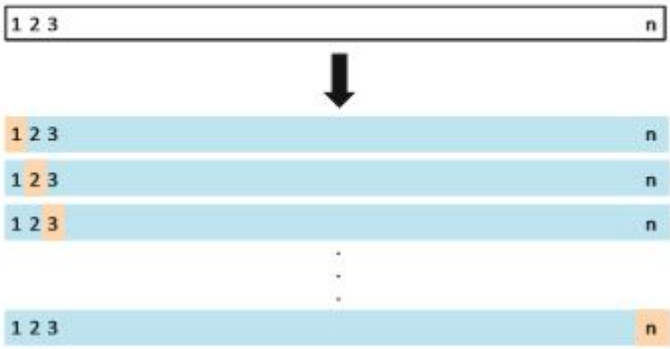

FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's.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

- validation set approach에서 발생한 단점을 줄이고자, train data의 양을 늘림
- 전체 데이터에서 하나의 데이터만 제외해 n-1개의 데이터로 모델을 fitting한 후, 하나의 데이터에 대해서 MSE를 구한다. 이 과정을 반복해서 얻는 n개의 MSE 평균을 Test MSE의 추정치로 사용

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

- 장점
  1. 데이터를 반만 사용하던 validation set approach와 달리 n-1개의 데이터를 사용해 모델을 fitting하기 때문에 bias가 적다. 즉 test MSE를 overestimate하지 않는다
  2. 추정치n개의 값에 대한 평균을 test MSE의 추정치로 사용하기 때문에 Training/ validation 구분에 따라 추정치가 달라질 일이 없다
  3. OLS 회귀분석에서는 n번의 과정을 반복할 필요없이 아래의 식을 통해 한번에 test MSE의 추정치를 구할 수 있다

  $$CV(n) = \frac{1}{N}\sum_{i=1}^{N}(\frac{y_i - \hat{y_i}}{1 - h_{ii}})^2$$

  증명은 뒷부분 참고!

- 단점
  1. 일반적으로는 n번을 반복해서 fitting을 해야하기 때문에 n이 커지면 fitting을 하는데 시간이 오래 걸린다
     - 그러면 n번 반복하지 말고 k번 반복하자 : **K-fold cross validation**
  2. LOOCV는 n-1개의 데이터로 모델을 fitting하기 때문에 bias가 낮지만 높은 variance를 가지고 있다.
     - 왜? 1개의 데이터를 제외하고 fitting을 해서 생성된 n개의 모델들은 서로 높은 correlation을 가지고 있기 때문에

1.3 K-Fold Cross Validation : K개의 그룹으로 나누자

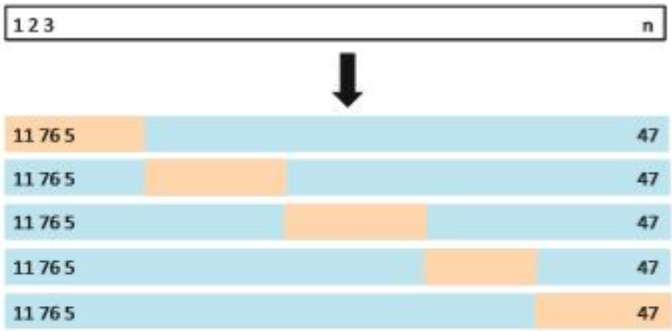

FIGURE 5.5. *A schematic display of 5-fold CV. A set of n observations is randomly split into five non-overlapping groups. Each of these fifths acts as a validation set (shown in beige), and the remainder as a training set (shown in blue). The test error is estimated by averaging the five resulting MSE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

chapters. The magic formula (5.2) does not hold in general, in which case the model has to be refit n times.

- 전체 데이터를 k개의 그룹으로 랜덤하게 나누고, k-1개의 그룹을 통해 모델을 fitting하고 그 나머지 그룹을 validation set으로 활용하는 방법
- k개의 MSE를 구해서 그 평균값을 Test MSE의 추정치로 사용한다
- K값은 주로 5나 10을 사용한다고 한다

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

- 장점
  1. LOOCV보다 연산속도가 더 빠르다
  2. Test MSE 추정치의 분산이 더 줄어든다 : bias - variance trade off 관계
     - 이러한 점에서 LOOCV보다 더 개선된 방법이라고 볼 수 있다

1.4 Bias - variance Trade Off

- bias: Training set의 데이터가 많을수록 모델의 bias가 줄어든다
  - the validation approach는 반개의 데이터만 활용해서 bias가 큰 반면에 LOOCV와 K-fold cross validations는 상대적으로 작은 bias를 갖는다
- Variance: LOOCV가 하나의 데이터만 제외하고 fitting을 한 것이기 때문에 n개의 fitting 결과가 거의 동일하다.
  - 수학적으로 이해하면 Variance를 구할 때 MSE의 공분산 값이 포함되는데, correlation이 높아지면 분산이 커짐

$$Var(CV(k)) = VAR(\frac{1}{k}MSE_1 + \frac{1}{k}MSE_2 + \ldots + \frac{1}{k}MSE_K)$$

$$= \frac{1}{k^2}[VAR(MSE_1) + VAR(MSE_2) + \ldots + VAR(MSE_K) + \ldots + \sum_{i=/=j} COV(MSE_i, MSE_J)]$$

    - 다른 데이터 셋으로 Loocv를 구하면 값이 많이 달라진다.

2. BootStrap

- Bootstrap은 통계학에서 복원추출을 의미. 우리가 가지고 있는 데이터에서 복원추출을 하여 여러개의 샘플을 만드는 방법
- 우리가 모수 θ에 대한 추정을 하고자 할 때, $\hat{\theta}$ 에 대한 분포를 모른다고 하자. 이 때 데이터에서 복원추출을 하여 여러 개의 sample을 만들고 각 sample에서 $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_n$을 생성해 bootstrap분포를 구할 수 있다. 이렇게 구한 bootstrap 분포가 모집단의 분포에 근사하게 된다(단 각각의 값이 iid 만족해야 함)

[예시]

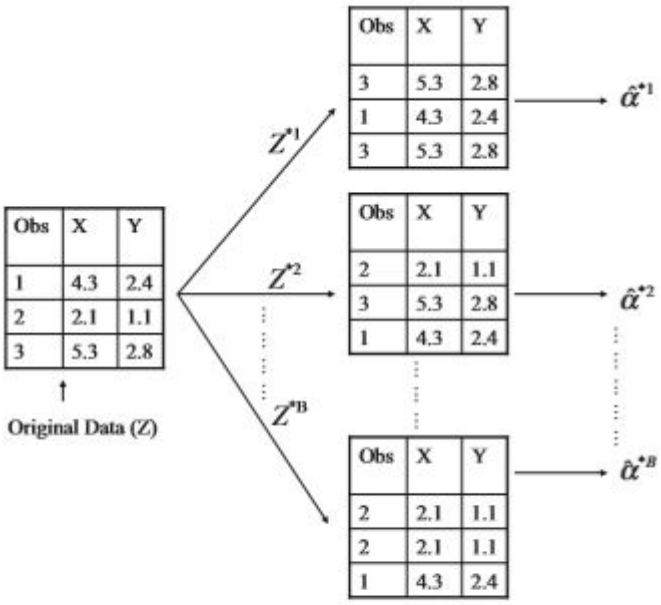

FIGURE 5.11. *A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.*

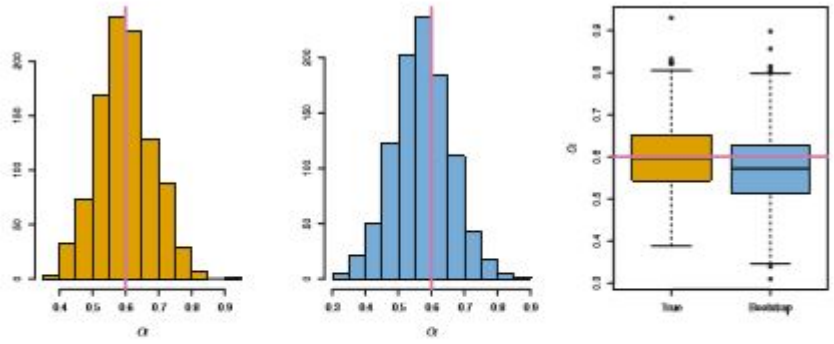

FIGURE 5.10. Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels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

- 1000개의 bootstrap sample을 뽑아 그 분포를 구했더니, 모분포에 근사한다는 것을 확인할 수 있다 > 부트스트랩은 매우 유용!!

lab 실습: validation set approach / k-fold cross validation

- https://github.com/a-martyn/ISL-python/tree/master/Notebooks (https://github.com/a-martyn/ISL-python/tree/master/Notebooks) 참조

In [ ]:
```
# 패키지 불러오기
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix
```

In [ ]:
```
# confusion_table, total_error_rate을 계산하기 위한 함수 정의
def confusion_table(confusion_mtx):
    """Renders a nice confusion table with labels"""
    confusion_df = pd.DataFrame({'y_pred=0': np.append(confusion_mtx[:, 0], confusion_mtx.sum(axis=0)[0]),
                                 'y_pred=1': np.append(confusion_mtx[:, 1], confusion_mtx.sum(axis=0)[1]),
                                 'Total': np.append(confusion_mtx.sum(axis=1), ''),
                                 '': ['y=0', 'y=1', 'Total']}).set_index('')
    return confusion_df

def total_error_rate(confusion_matrix):
    """Derive total error rate from confusion matrix"""
    return 1 - np.trace(confusion_mtx) / np.sum(confusion_mtx)
```

In [ ]:
```
# 데이터 불러오기
default_df = pd.read_csv('./data/Default.csv', index_col='Unnamed: 0')
default_df = default_df.reset_index().drop('index', axis=1)

# 카테고리 변수를 위해 dummies 사용, drop은 각 변수 당 생기는 두개의 컬럼 중 하나를 제거하기 위함
default_df = pd.get_dummies(default_df, dtype=np.float64).drop(['default_No', 'student_No'], axis=1)

display(default_df)
```

여기서 문제!

1. 다음의 순서에 따라 validation set approach를 이용해서 test error를 추정해라

데이터는 /data폴더에 저장되어 있다

1. 데이터를 랜덤으로 training set과 validation set으로 나눠라
2. training set을 이용해서 multiple logistic model로 fitting해라
3. 위에 정의된 함수를 이용해서 confusion matrix와 total_error_rate을 구해라

In [ ]:
```
# 1. 데이터를 training set과 validation set으로 나눠라

np.random.seed()
train = np.random.rand(len(default_df)) < 0.5

response   = 'default_Yes'
predictors = ['income', 'balance', 'student_Yes']

## fill this blank!
```

In [ ]:
```
# 2. training set을 이용해서 multiple logistic model로 fitting해라
# 힌트: LogisticRegression(), l.fit(x,y)

## fill this blank!
```

In [ ]:
```
# 3. 위에 정의된 함수를 이용해서 confusion matrix와 total_error_rate을 구해라

##fill this blank!
```

2. 이번에는 k-fold cross validation을 이용해 k=5일 때 모델의 accuracy를 구해라

힌트 https://homeproject.tistory.com/6 (https://homeproject.tistory.com/6)

In [ ]:
```
from sklearn.model_selection import cross_val_score
from sklearn.linear_model import LogisticRegression
```

In [ ]:
```
logit = LogisticRegression()

response   = 'default_Yes'
predictors = ['income', 'balance', 'student_Yes']


## fill this blank!
```

증명내용

$$\sum_{i=1}^{n}(y_i-\hat{y}_{i(i)})^2=\sum_{i=1}^{n}\left(\frac{y_i-\hat{y}_i}{1-h_{ii}}\right)^2$$

- $\hat{y}_{i(i)}$: i번째 data를 제외하고 fitting한 모델에 x_i를 넣어 구한 예측값

h_{ii} : H matrix의 (i,i)값

$= X(X^TX)^{-1}X^T$

Show $y_i-\hat{y}_{i(i)}=\dfrac{y_i-\hat{y}_i}{1-h_{ii}}$

1. $f(\underline{x})=\beta_0+\beta_1\underline{x_1}+\cdots+\beta_{p-1}\underline{x_{p-1}}\quad\cdots(A)$

Data : $(\underline{x_1},y_1)\ (\underline{x_2},y_2)\ \cdots\ (\underline{x_n},y_n)$

2. LSE gives $\hat{f}$: $\arg\min_f\left\{\sum_{i=1}^{n}(y_i-f(\underline{x_i}))^2\right\}=Hy\ \cdots(B)$

$\therefore\ \sum_{i=1}^{n}(y_i-\hat{f}(\underline{x_i}))^2\le\sum_{i=1}^{n}(y_i-f(x_i))^2$ for $\forall f$ form of (A)

3. k^{th} data를 제외하고 fitting 한 모델 $\rightarrow \hat{f}^{(k)}$

$\hat{f}^{(k)}(\underline{x_k})=\hat{y}_{k(k)}$

$$\sum_{i\neq k}^{n}(y_i-\hat{f}^{(k)}(x_i))^2+\underbrace{(\hat{y}_{k(k)}-\hat{f}^{(k)}(x_k))^2}_{=0}$$

$$\le\sum_{i\neq k}^{n}(y_i-f^{(k)}(x_i))^2$$

$$\le\sum_{i\neq k}^{n}(y_i-f^{(k)}(x_i))^2+\underbrace{(\hat{y}_{k(k)}-f^{(k)}(x_k))^2}_{\ge 0}$$

$$\therefore \sum_{i\neq k}^{n}(y_i-\hat{f}^{(k)}(x_i))^2+(\hat{y}_{k(k)}-\hat{f}^{(k)}(x_k))^2$$

$$\leq \sum_{i\neq k}^{n}(y_i-f^{(k)}(x_i))^2+(\hat{y}_{k(k)}-f^{(k)}(x_k))^2 \cdots (c)$$

→ $\hat{f}^{(k)}(x_i)$ is the LSE on the data with the k^{th} case (x_k, y_k) replaced with $(x_k, \hat{y}_{k(k)})$

4. $\tilde{y} = (y_1 \cdots y_{k(k)} \cdots y_n)'$

(c) $\sum_{i=1}^{n}(\tilde{y}-\hat{f}^{(k)}(x_i))^2 \leq \sum_{i=1}^{n}(\tilde{y}-f^{(k)}(x_i))^2$

$\hat{f}^{(k)} = H\tilde{y}$

5. $\hat{f}(x_k)-\hat{f}^{(k)}(x_k) = [H(y-\tilde{y})]_k$

full data 사용

$$= \begin{bmatrix} h_{11} & \cdots & h_{1n} \\ & & \\ h_{n1} & \cdots & h_{nn} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ y_k-\hat{y}_{k(k)} \\ \vdots \\ 0 \end{bmatrix} = h_{kk}(y_k-\hat{y}_{k(k)})$$

$\Rightarrow \hat{y}_k-\hat{y}_{k(k)} = h_{kk}(y_k-\hat{y}_{k(k)})$

식 정리하면

$$\sum_{i=1}^{n}(y_i-\hat{y}_{i(i)})^2 = \sum_{i=1}^{n}\left(\frac{y_i-\hat{y}_i}{1-h_{ii}}\right)^2$$