15) Cross-Validation

Vitor Kamada

February 2018

Tables, Graphics, and Figures from

An Introduction to Statistical Learning

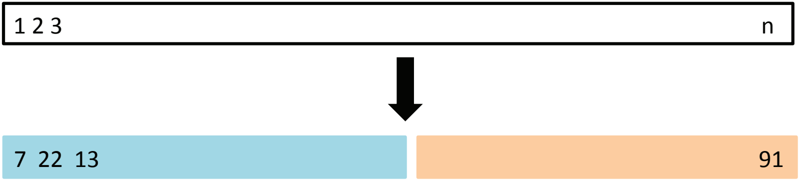
James et al. (2017): Chapters: 5.1, 5.3.1, 5.3.2, 5.3.3

The Elements of Statistical Learning

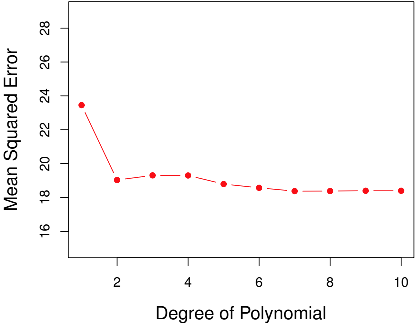
Hastie et al. (2017): Chapter: 7.10

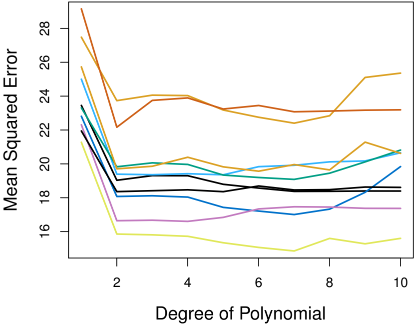
Training Set vs Validation or Hold-out Set

Randomly division in two part:

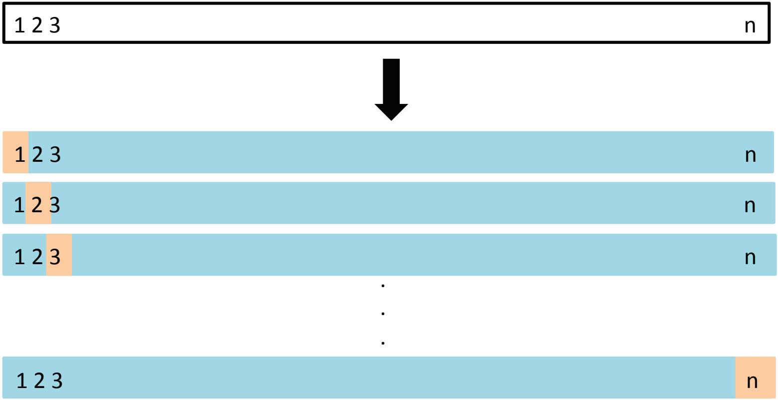


Random Split (10x)





Leave-One-Out Cross-Validation (LOOCV)



LOOCV

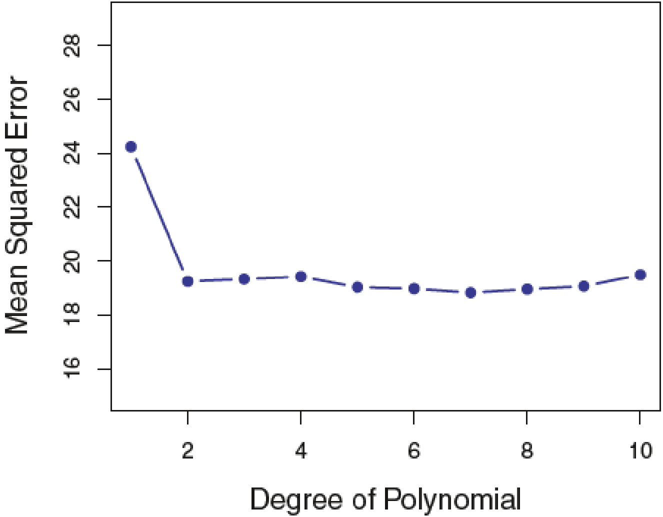
$$\{(x_2, y_2), \ldots, (x_n, y_n)\} \rightarrow \text{Training Set}$$

$$(x_1, y_1) \rightarrow \text{Validation Set}$$

$$MSE_1 = (y_1 - \hat{y}_1)^2$$

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

LOOCV: mpg on hp ...



Generalized Cross-Validation (GCV)

$$\hat{y} = Sy$$

$$\frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{f}^{-i}(x_i)]^2 = \frac{1}{N}\sum_{i=1}^{N}[\frac{y_i - \hat{f}(x_i)}{1 - S_{ii}}]^2$$

$$GCV(\hat{f}) \cong \frac{1}{N}\sum_{i=1}^{N}[\frac{y_i - \hat{f}(x_i)}{1 - \frac{trace(S)}{N}}]^2$$

$$trace(S)\text{: Effective \# of Parameters}$$

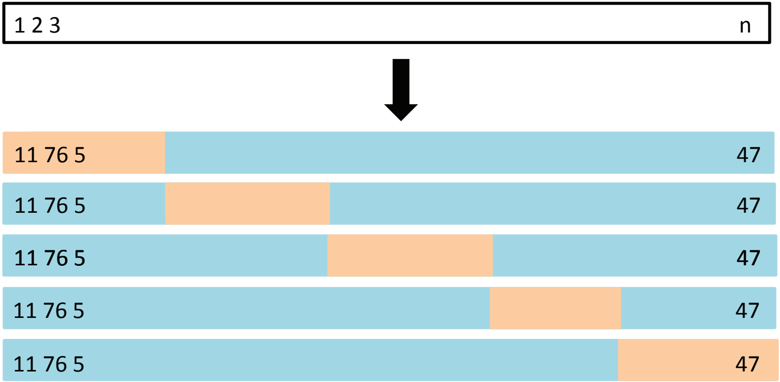
LOOCV for OLS

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

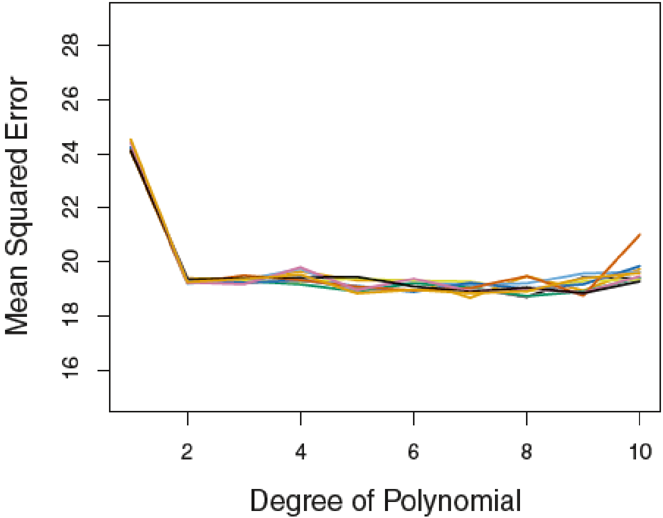
$$h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

k-Fold Cross-Validation

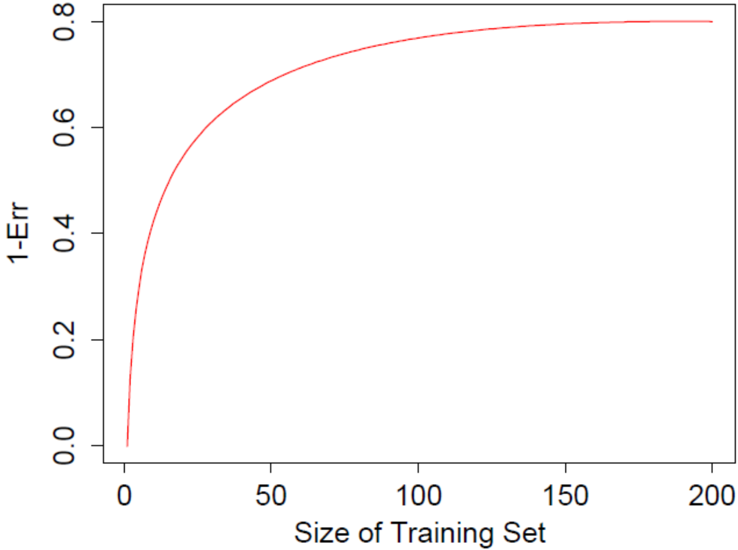
$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^{k} MSE_i$$



10-fold CV: mpg on hp ...



Hypothetical Learning Curve for a Classifier



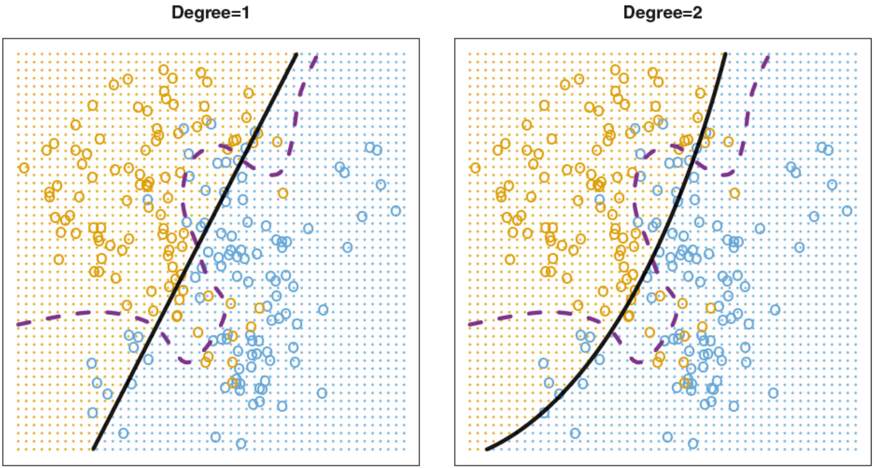
Cross-Validation on Classification Problems

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

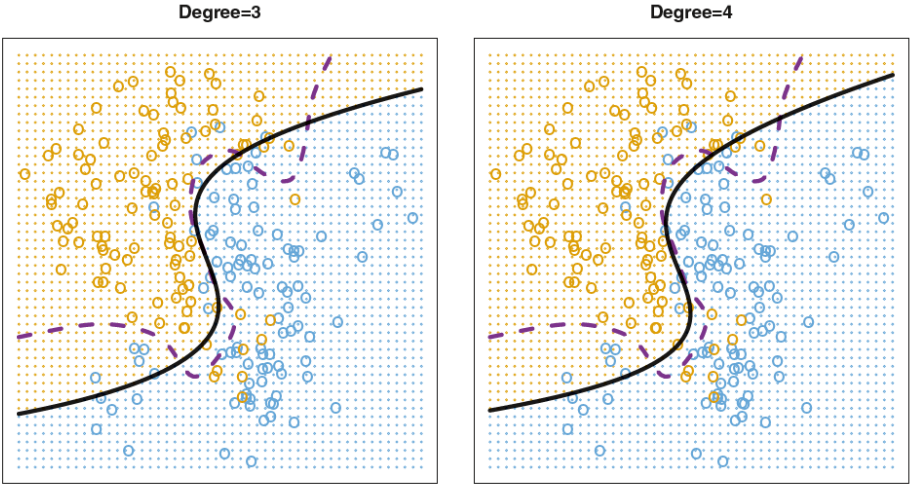
$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2$$

Test Error Rates: 20.1% and 19.7%

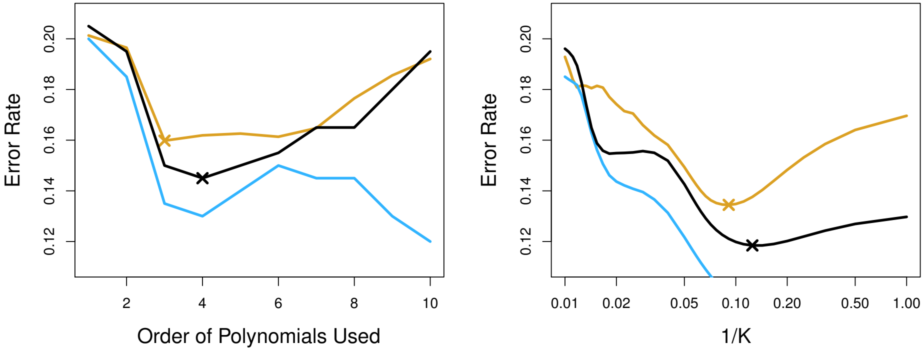
Bayes Error Rate: 13.3%



Test Error Rates: 16% and 16.2%



Test (brown), Training (blue), and 10-fold CV Error (black)



attach(Auto); set.seed(1); train=sample(392,196)

```
lm.fit=lm(mpg~horsepower,data=Auto,subset=train)
mean((mpg-predict(lm.fit,Auto))[-train]^2)
```

26.14

```
lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
mean((mpg-predict(lm.fit2,Auto))[-train]^2)
```

19.82

```
lm.fit3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
mean((mpg-predict(lm.fit3,Auto))[-train]^2)
```

19.78

set.seed(2); train=sample(392,196)

```
lm.fit=lm(mpg~horsepower,subset=train)
mean((mpg-predict(lm.fit,Auto))[-train]^2)
```

23.3

```
lm.fit2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
mean((mpg-predict(lm.fit2,Auto))[-train]^2)
```

18.9

```
lm.fit3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
mean((mpg-predict(lm.fit3,Auto))[-train]^2)
```

19.3

LOOCV in R

```
library(boot); lm.fit=glm(mpg~horsepower,data=Auto)
cv.err=cv.glm(Auto,glm.fit); cv.err$delta
```

24.23151 24.23114

```
cv.error=rep(0,5)
for (i in 1:5){
 glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
 cv.error[i]=cv.glm(Auto,glm.fit)$delta[1] }
cv.error
```

24.23 19.25 19.33 19.42 19.03

k-Fold Cross-Validation in R

```
set.seed(17); cv.error.10=rep(0,10)
for (i in 1:10){
glm.fit=glm(mpg~poly(horsepower,i),data=Auto)
cv.error.10[i]=cv.glm(Auto,glm.fit,K=10)$delta[1]
}
cv.error.10
```

24.20 19.19 19.3 19.34 18.88

19.02 18.9 19.71 18.95 19.50