

9) K-Nearest Neighbors

Vitor Kamada

January 2018

Tables, Graphics, and Figures from

James et al. (2017): Chapters: 2.2.3, 3.5, 4.5,
4.6.5, 4.6.6

Hastie et al. (2017): Chapters: 2.3

Training Error and Test Error

$$Y = f(X) + \epsilon$$

$$\hat{Y} = \hat{f}(X)$$

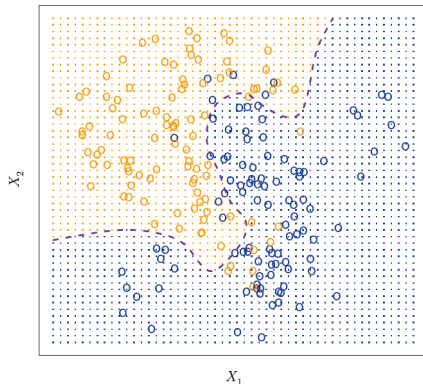
$$\{(x_1, y_1), \dots, (x_n, y_n)\}$$

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

$$\text{Ave}(I(y_0 \neq \hat{y}_0))$$

Bayes Classifier and Error Rate

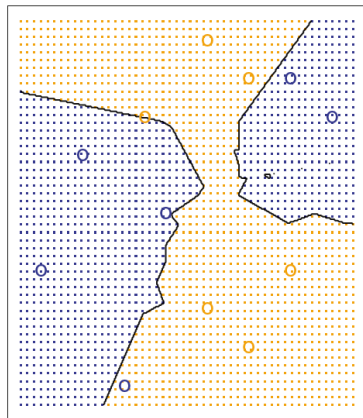
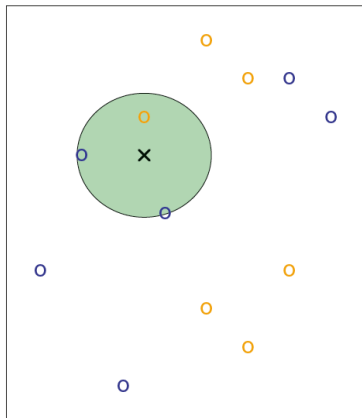
$$Pr(Y = j|X = x_0)$$



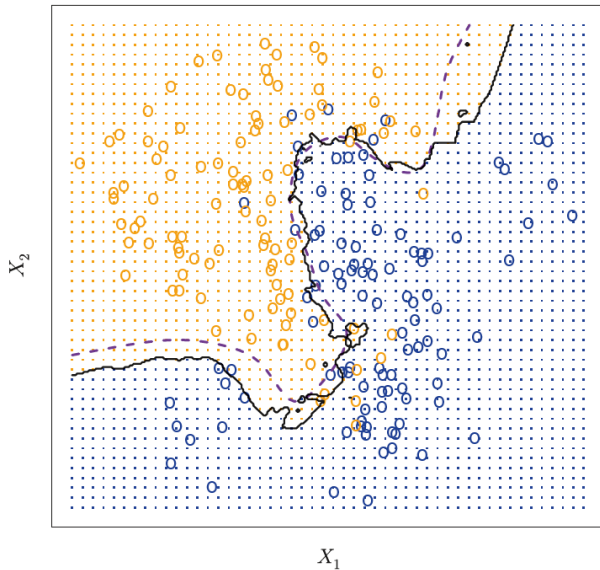
$$1 - E[\max_j Pr(Y = j|X)] = 0.13$$

K-Nearest Neighbors (K=3)

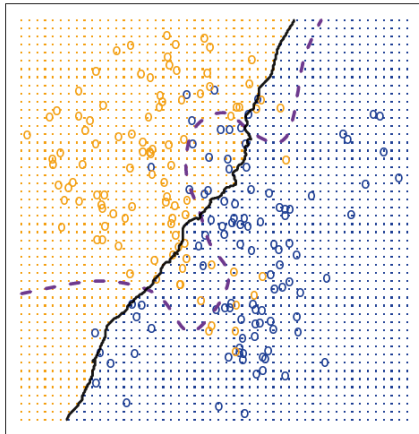
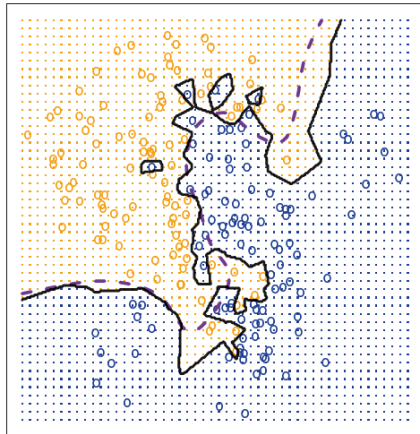
$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$



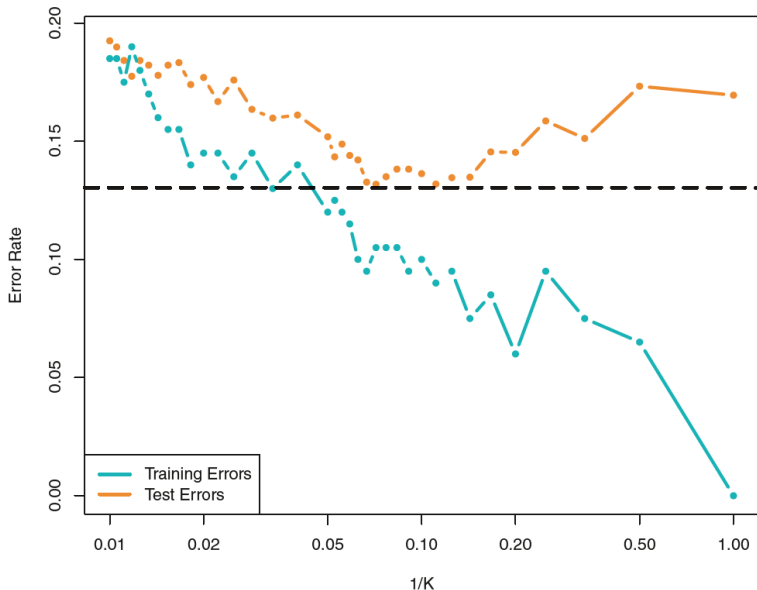
KNN: K=10



KNN: $K=1$ and $K=100$

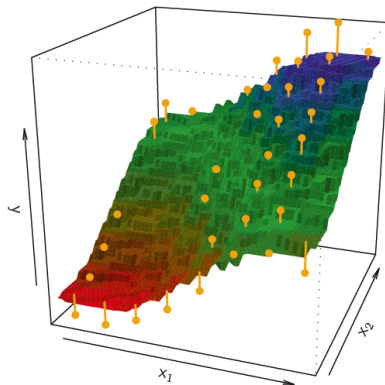
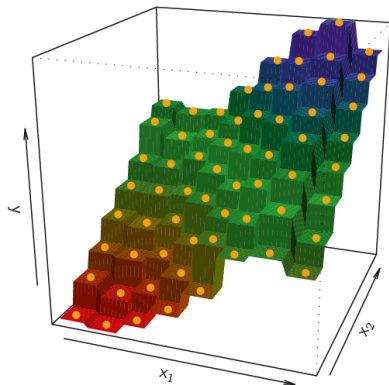


KNN Training and Test Error Rate

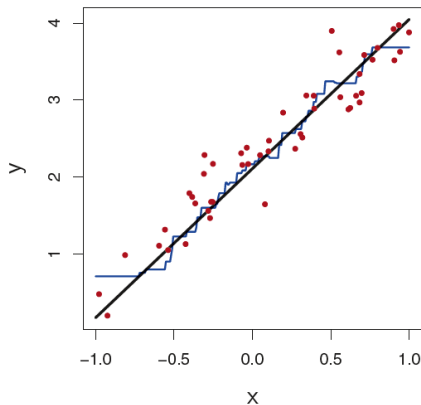
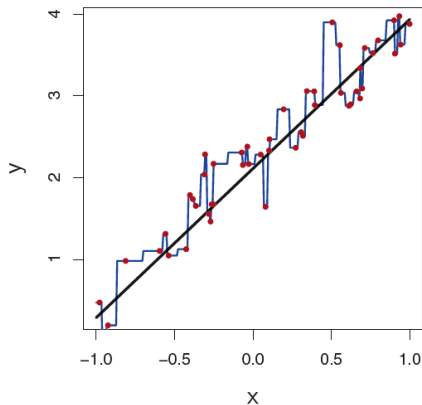


KNN Regression: K=1 and K=9

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in N_0} y_i$$



One-dimension KNN Regression: $K=1$ and $K=9$



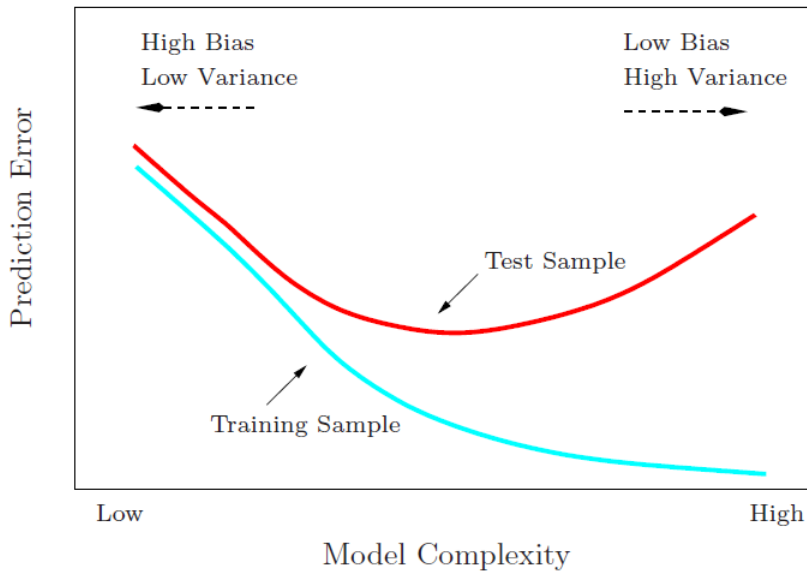
Mean Squared Error (MSE)

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ &= E[f(X) - \hat{f}(X)]^2 + \text{Var}(\epsilon) \end{aligned}$$

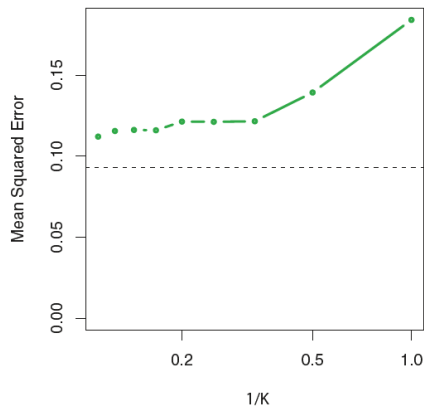
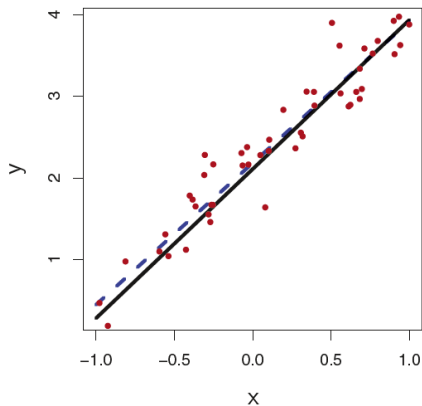
$$MSE(x_0) = E_{\tau}[f(x_0) - \hat{y}_0]^2$$

$$\begin{aligned} E_{\tau}[\hat{y}_0 - E_{\tau}(\hat{y}_0)]^2 &+ [E_{\tau}(\hat{y}_0) - f(x_0)]^2 \\ \text{Var}_{\tau}(\hat{y}_0) &+ [\text{Bias}(\hat{y}_0)]^2 \end{aligned}$$

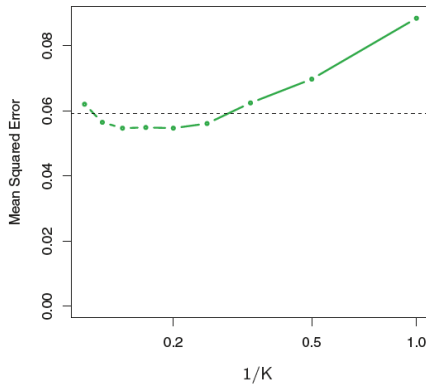
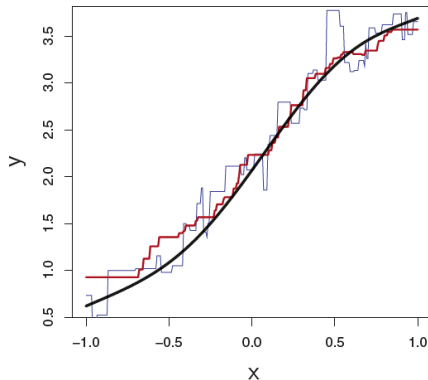
Bias-Variance Tradeoff



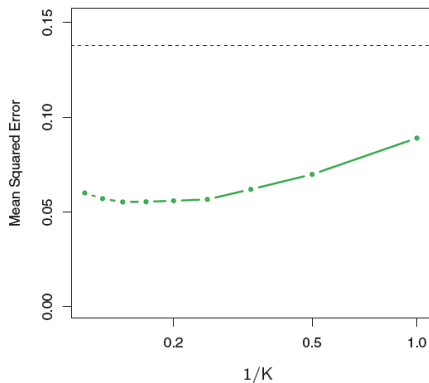
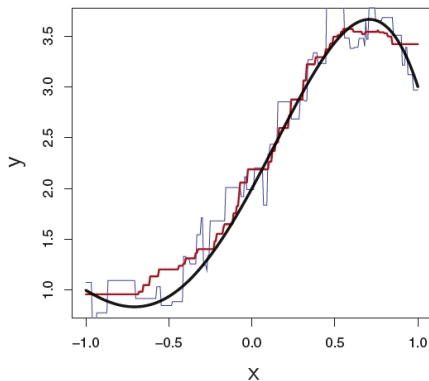
MSE: OLS vs KNN



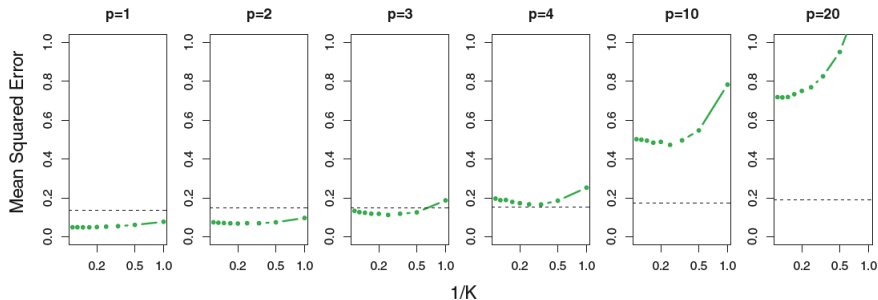
Slightly Non-Linear Relationship



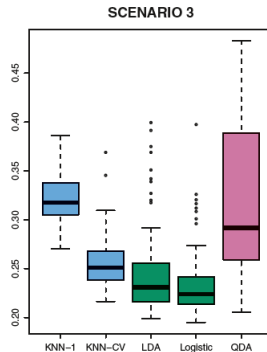
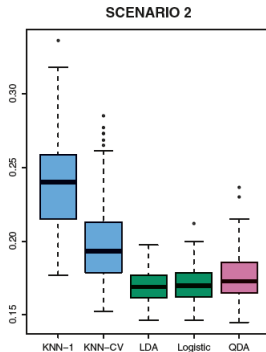
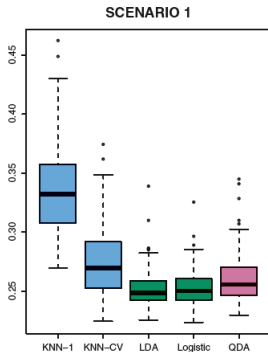
Strongly Non-Linear Relationship



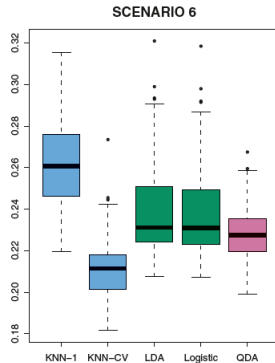
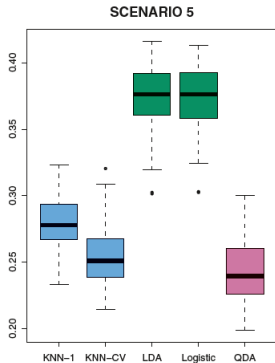
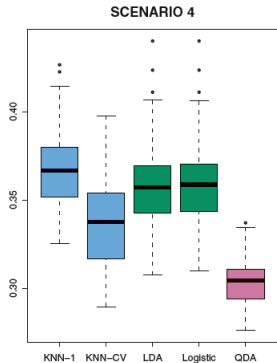
Additional Noise Variables



Test Error Rates: Linear Scenarios



Test Error Rates: Non-Linear Scenarios



Caravan Insurance Data

```
library(ISLR); library(class)  
dim(Caravan)
```

5822 86

```
summary(Purchase)
```

No	Yes
5474	348

$$\frac{348}{5822} \approx 6\%$$

Standardize the Data

```
standardized.X=scale(Caravan[, -86])
```

```
var(Caravan[,1])
```

165

```
var(Caravan[,2])
```

0.165

```
var(standardized.X[,1])
```

1

```
var(standardized.X[,2])
```

1

K = 1

```
set.seed(1); test=1:1000  
train.X=standardized.X[-test,]  
test.X=standardized.X[test,]  
train.Y=Purchase[-test]  
test.Y=Purchase[test]  
knn.pred=knn(train.X,test.X,train.Y,k=1)  
mean(test.Y!=knn.pred) 0.118  
mean(test.Y!="No") 0.059
```

K = 3 and K=5

```
knn.pred=knn(train.X,test.X,train.Y,k=3)
```

```
table(knn.pred,test.Y)
```

knn.pred/test.Y	No	Yes
No	920	54
Yes	21	5

$$\frac{5}{26} = 19.2\%$$

```
knn.pred=knn(train.X,test.X,train.Y,k=5)
```

```
table(knn.pred,test.Y)
```

knn.pred/test.Y	No	Yes
No	930	55
Yes	11	4

$$\frac{4}{15} = 26.7\%$$

Logistic Regression

```
glm.fit=glm(Purchase~.,data=Caravan,family=binomial,  
            subset=-test)  
glm.probs=predict(glm.fit,Caravan[test,],type="response")  
glm.pred=rep("No",1000)  
glm.pred[glm.probs>.5]="Yes"  
table(glm.pred,test.Y)
```

glm.pred/test.Y	No	Yes
No	934	59
Yes	7	0

Cut-off of 0.25

```
glm.pred=rep("No",1000)  
glm.pred[glm.probs>.25]="Yes"  
table(glm.pred,test.Y)
```

glm.pred/test.Y	No	Yes
No	919	48
Yes	22	11

$$\frac{11}{33} = 33\%$$