Bruce Campbell ST-617 HW 1

Tue Jun 28 22:10:44 2016

Chapter 2

Problem 7

```r
library("knitr")
library("pander")
X = matrix(data = c(0, 2, 0, 0, -1, 1, 0, 3, 0, 1, 1, 0, 1, 0, 0, 0, 3, 2, 1,
    1, 0), nrow = 7, ncol = 3)

pander(X)
```

0	3	0
2	0	0
0	1	3
0	1	2
-1	0	1
1	1	1
0	0	0

```r
D <- as.matrix(dist(X, method = "euclidean", diag = FALSE, upper = FALSE, p = 2))

# Adding class labels
DF <- cbind(X, c("Red", "Red", "Red", "Green", "Green", "Green", "UNK"))
```

This is the distance matrix

```r
pander(D)
```

1	2	3	4	5	6	7
0	3.606	3.606	2.828	3.317	2.449	3
3.606	0	3.742	3	3.162	1.732	2
3.606	3.742	0	1	2.449	2.236	3.162
2.828	3	1	0	1.732	1.414	2.236
3.317	3.162	2.449	1.732	0	2.236	1.414
2.449	1.732	2.236	1.414	2.236	0	1.732
3	2	3.162	2.236	1.414	1.732	0

And the distances from the test points to the training points is

```
testDist <- D[7, -7]
pander(testDist)
```

1	2	3	4	5	6
3	2	3.162	2.236	1.414	1.732

When K=1 the distance to the nearest neighbor is 1.4142136 - note we removed the test point.

```
index <- which.min(D[7, -7])
classLabel <- (DF[index, 4])
```

a)

The predicted label for K=1 is Green

The KNN classifier estimates the class conditional probability using K nearest neighbors as

$$P(Y = color|X = x_0) = \frac{1}{k}\sum_{N_k} I(y_i == color)$$

for K=1 this reduces to setting the color to that of the nearest neighbor - which is green.

When K=3 we have

```
Z <- sort(testDist, index.return = TRUE)

class1stNearest <- Z$ix[1]
class2dnNearest <- Z$ix[2]
class3rdNearest <- Z$ix[3]
```

b)

For K=3 the classes of the three nearest neighbors are

```
pander(c(DF[class3rdNearest, 4], DF[class2dnNearest, 4], DF[class3rdNearest,
    4]))
```

Red, *Green* and *Red*

And

$$P(Y = red|X = (0,0,0)) = \frac{1}{3}2) = \frac{2}{3}$$

$$P(Y = green|X = (0,0,0)) = \frac{1}{3}1) = \frac{1}{3}$$

So KNN with K=3 classifies the test point as red.

c)

If the optimal decision boundary is highly non-linear we would expect the best value of K to be a small number. This is because local information is lost when including large number of points. When K gets very large we average over a large area and the optimal decision boundary is effectively smoothed out. When K is small we classify based on local information and we expect KNN to preform better for highly non linear boundaries.