Doug Woodward

CS613 HW 3

|  |  |
| --- | --- |
| # | Answer |
| 1 | 1. Using d(x,y) = sqrt((x0-y0)^2+(x1-y01)^2+(x2-y2)^2)    1. 3    2. 2    3. sqrt(10)    4. sqrt(5)    5. sqrt(2)    6. sqrt(3) 2. Green, as point 5 is the closest 3. Red, as the three closest points are 5, 6, 2, the latter two being Red. 4. If the Bayes decision Boundary is highly nonlinear we would want to use a smaller value of K, as with smaller values of K, a more complex Bayes boundary is drawn by KNN, as the most influential points are the local K, whereas a high value of K, produces a closer to linear boundary. |
| 2 | 1. If x is between .5 and .95, then we can know that we know that on average, .1\*n will be within the interval [x-.05,x+.05], having length .1, therefore 10% for this case. When 0<x<.5, we use an interval of [0, x+.05], similarly, when .95>x>1 we use [x-.05, 1]. To compute the average, we must take the integral of these combined intervals, which produces an average of 9.75% 2. From the result above, we can say that for both X1 and X2, the average is 9.75%, therefore overall, 9.75%\*9.75% = 9.5% 3. Again, we can use the above to find that on average we are using 9.75%^100 of the available variables, which is effectively 0. 4. Given the result from c, we can say as soon as dimensionality increases above very small amounts, there are very few observations near each other. Mathematically, we can say    1. limit(p -> infinity) 9.75%^p = 0 5. p = 1 -> 0.10^1 = .1, p = 2 -> 0.10^½ = 0.316 , p = 3 -> 0.10^1/100 = .977    1. This shows that as p increases we must include more and more o the full dataset to achieve the given hypercube, with p = 100 we need to include 97% of observations to fit this criteria, showing that scaling knn with dimensions is far from ideal |
| 3 | ID Money Free For Gambling Fun Machne Learnin Spam Euclid Dis  1 3 0 0 0 0 0 0 TRUE 3.605551275 2 1 2 1 1 1 0 0 TRUE 2.449489743 3 0 0 1 1 1 0 0 TRUE 2.236067977 4 0 0 1 0 3 1 1 FALSE 3.16227766 5 0 1 0 0 0 1 1 FALSE 1  0 1 1 0 0 1 1 0   1. Not Spam - as #5 is closest 2. Spam - as #5, #2, #3 are the three closest, with 2,3 being Spam 3. NA 4. Manhattan Dis  2.645751311 2.449489743 2.236067977 2 1 0   The Manhattan distance with k = 3 results in Not Spam |
| 4 | 1. 4.589133 3. 5.968967 4. The non weighted actually gives better results here, likely because of the small sample size. |

# read data from csv  
rm(list=ls())  
data = read.csv("cpi.csv")  
set.seed(42)  
  
normalize <- function(x) {  
 return( (x-min(x))/(max(x) - min(x)))  
}  
denormalize <- function(x,minval,maxval) {  
Press ENTER or type command to continueal)  
   
minvec <- sapply(data[,2:(ncol(data)-1)], min)  
maxvec <- sapply(data[,2:(ncol(data)-1)], max)  
  
data[,2:(ncol(data)-1)] = apply(data[,2:(ncol(data)-1)],2, normalize)  
russia <- data[nrow(data),]  
data = data[-nrow(data),]  
  
  
russia.trim = russia[,2:(ncol(russia)-1)]  
dists = apply(data[2:(ncol(data)-1)],1,function(x)sqrt(sum((x-russia.trim)^2)))  
dists = sort(dists)  
dists  
dist.indexes = names(dists)  
dist.indexes  
data[,2:(ncol(data)-1)] = as.data.frame(Map(denormalize, data[,2:(ncol(data)-1)], minvec, maxvec))  
data[dist.indexes[1],]  
data[dist.indexes[2],]  
data[dist.indexes[3],]  
  
mean(c(data[dist.indexes[1],ncol(data)],  
 data[dist.indexes[2],ncol(data)],  
 data[dist.indexes[3],ncol(data)]))