Week 11: Machine Learning

Yograj Karki

6/5/2021

Loading packages and working directories

```
#Packages
library(ggplot2)
library(plyr)
library(dplyr)
library(tidyverse)
library(factoextra)
library(cluster)
library(caret)
library(class)

  setwd("~/MSDS/DSC520/dsc520")
```

Loading the data

```
binary_df <- read.csv("data/binary-classifier-data.csv")
trinary_df <- read.csv("data/trinary-classifier-data.csv")

binary_df$label <- as.factor(binary_df$label)
trinary_df$label <- as.factor(trinary_df$label)

head(binary_df)
```

```
##   label        x        y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

```
head(trinary_df)
```

```
##   label        x        y
## 1     0 30.08387 39.63094
```

```
## 2      0 31.27613 51.77511
## 3      0 34.12138 49.27575
## 4      0 32.58222 41.23300
## 5      0 34.65069 45.47956
## 6      0 33.80513 44.24656
```

Scatter plot binary classifier

```r
library(ggvis)
```

```
## Warning: package 'ggvis' was built under R version 3.6.2
```

```
##
## Attaching package: 'ggvis'
```

```
## The following object is masked from 'package:ggplot2':
##
##     resolution
```

```r
binary_df %>% ggvis(~x, ~y, fill = ~label) %>% layer_points()
```

Scatter plot trinary classifier data

```r
trinary_df %>% ggvis(~x, ~y, fill = ~label) %>% layer_points()
```

k nearest neighbors algorithm

Data preparation

```r
# Generate a random number that is 80% of the total number of rows in data set.
set.seed(42)
# Random sampling
random_binary <- sample(1:nrow(binary_df), 0.8 * nrow(binary_df))
random_trinary <- sample(1:nrow(trinary_df), 0.8*nrow(trinary_df))
```

Splitting training and testing data

```r
#Binary
train_binary <- binary_df[random_binary,]
test_binary <- binary_df[-random_binary,]
#Trinary
train_trinary <- trinary_df[random_trinary,]
```

```r
test_trinary <- trinary_df[-random_trinary,]


# Creating separate dataframe for our target
train_binary_labels <- binary_df[random_binary,1]
test_binary_labels <-binary_df[-random_binary,1]
train_trinary_labels <- trinary_df[random_trinary,1]
test_trinary_labels <-trinary_df[-random_trinary,1]
```

Fit a k nearest neighbors' model for each dataset for k=3, k=5, k=10, k=15, k=20, and k=25. Compute the accuracy of the resulting models for each value of k. Plot the results in a graph where the x-axis is the different values of k and the y-axis is the accuracy of the model.

Confusion matrix formula

```r
# Formula for Binary confusion matrix
my.statistics <- function(Actual,Predicted) {
  confusion.table <- table(Actual=Actual,Predicted=Predicted)
  output <- list(confusion.table=confusion.table)
  TN <- confusion.table[1]
  FN <- confusion.table[2]
  FP <- confusion.table[3]
  TP <- confusion.table[4]
  output$accuracy <- (TP+TN)/sum(confusion.table)
  output$precision <- (TP)/(TP+FP)
  output$sensitivity <- (TP)/(TP+FN)
  output$specificity <- (TN)/(TN+FP)
  output$FPR <- (FP)/(TN+FP)

  return(output)
}
```

```r
# k = 3 on Binary data
knn_pred <- knn(train_binary[,c("x","y")],
                test_binary[,c("x", "y")],
                cl = train_binary$label,
                k = 15)

# Accuracy
my.statistics(test_binary$label, knn_pred)
```

```
## $confusion.table
##       Predicted
## Actual   0   1
##      0 153   5
##      1   5 137
##
## $accuracy
## [1] 0.9666667
```

```
##
## $precision
## [1] 0.9647887
##
## $sensitivity
## [1] 0.9647887
##
## $specificity
## [1] 0.9683544
##
## $FPR
## [1] 0.03164557
```

k = 5

k = 10

```r
# k = 10 on Binary data
knn_pred <- knn(train_binary[,c("x","y")],
                test_binary[,c("x", "y")],
                cl = train_binary$label,
                k = 10)

# Accuracy
my.statistics(test_binary$label, knn_pred)$accuracy
```

```
## [1] 0.97
```

k = 15

```r
# k = 15 on Binary data
knn_pred <- knn(train_binary[,c("x","y")],
                test_binary[,c("x", "y")],
                cl = train_binary$label,
                k = 15)

# Accuracy
my.statistics(test_binary$label, knn_pred)$accuracy
```

```
## [1] 0.9666667
```

k = 20

```r
# k = 20 on Binary data
knn_pred <- knn(train_binary[,c("x","y")],
                test_binary[,c("x", "y")],
                cl = train_binary$label,
```

```
                k = 20)

# Accuracy
my.statistics(test_binary$label, knn_pred)$accuracy
```

```
## [1] 0.9566667
```

k = 25

```
# k = 25 on Binary data
knn_pred <- knn(train_binary[,c("x","y")],
                test_binary[,c("x", "y")],
                cl = train_binary$label,
                k = 25)

# Accuracy
my.statistics(test_binary$label, knn_pred)$accuracy
```

```
## [1] 0.9633333
```

KNN for Trinary classifier

k = 3

```
knn_pred <- knn(train_trinary[, c("x", "y")],
                test_trinary[,c("x","y")],
                cl = train_trinary$label,
                k = 3)

my.statistics(test_trinary$label, knn_pred)$accuracy
```

```
## [1] 0.2547771
```

k = 5

```
knn_pred <- knn(train_trinary[, c("x", "y")],
                test_trinary[,c("x","y")],
                cl = train_trinary$label,
                k = 5)

my.statistics(test_trinary$label, knn_pred)$accuracy
```

```
## [1] 0.2579618
```