kmeans

MA Mingyu 14110562D

4/4/2017

COMP4433 Assignment 2 Question 3a

derek.ma@connect.polyu.hk derek.ma

Set Up

Import data, delete first column, set initial cluster centers to first two records and set k is equal to 2.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#PREPARE DATA IN USE
data <- read.csv("~/Google Drive/_DM/2_Assignments/Ass2/data_q3.csv", stringsAsFactors = FALSE)
#delete first column for this specific case becasue it is not a data record
data <- data[,2:length(colnames(data))]
#SET VALUE OF K
k <- 2
#SET INITIAL CLUSTER CENTERS
centers <- data[0,]
colnames(centers) <- colnames(data)
for (i in 1:k){
  #set first two record as initial centers
  centers[i,] <- data[i,]
}
```

Basic Function 1 - Distance Function

In this case, Euclidean Distance is used to calculate the dissimilarities.

```
distance <- function(vector1, vector2){
  #Euclidean distance
  #Input: two vectors of data with same length
  #Input example: c(1,2,3); c(2,3,4)
  count <- 0
  for (i in 1:length(vector1)){
    count = count + (vector1[i] - vector2[i])^2
  }
  count^(1/2)
}
```

Basic Function 2 - Compare Similarity and Assign Objects to Clusters

In this function, each record can be decivded belong to which clusters.

```
assign <- function(objectsData, centersData){
  #OUTPUT a data frame with new cluster information
  #FOR EACH RECORDS
  result <- objectsData %>%
    mutate(cluster = NA)
  for (i in 1:nrow(objectsData)){
    whichCenter <- 0
    currentMinDist <- -1
    #COUNT DISSIMILARITY BETWEEN IT AND CENTERS
    for (j in 1:nrow(centersData)){
      distValue <- distance(
        as.numeric(objectsData[i,]),
        as.numeric(centersData[j,]))
      if (distValue < currentMinDist || whichCenter == 0){
        #FOUND CENTER WITH SMALLER DISSIMILARITY
        whichCenter <- j
        currentMinDist <- distValue
      }
    }
    #SET THIS CENTER AS CLUSTER
    result[i,"cluster"] <- whichCenter
  }
  result
}
```

Basic Function 3 - Calculate Mean Values of Objects and Update Centers

In this function, the centers will be updated to the mean of clustered objects.

```
update <- function(objectsData, centersData){
  #INPUT  objectsData: the data frame with original data and corresponding cluster information
  #INPUT  centersData: all last round data for all centers
  #OUTPUT a data frame with new centers
  #FOR EACH CENTER
  result <- centersData
  for (i in 1:nrow(centersData)){
    #GET ALL NODES IN THIS CLUSTER
    clusterData <- subset(objectsData, objectsData[,"cluster"] == i)
    #CALCULATE MEAN FOR EACH FEATURE & UPDATE CENTERS
    for (j in 1:ncol(centersData)){
      result[i,j] <- mean(clusterData[,j])
    }
  }
  result
}
```

First Round

Run the algorithm for the first time.

```
data1 <- assign(data, centers)
data1
```

```
##       B     C    D   E   F G cluster
## 1  16.9 4.360 2.73 155 350 8       1
## 2  15.5 4.054 2.26 142 351 8       2
## 3  30.0 2.155 3.70  68  98 4       2
## 4  30.9 2.230 3.37  75 105 4       2
## 5  20.6 3.380 2.73 105 231 6       2
## 6  20.8 3.070 3.08  85 200 6       2
## 7  18.1 3.410 2.73 120 258 6       2
## 8  16.5 3.955 2.26 138 351 8       2
## 9  35.1 1.915 2.97  80  98 4       2
## 10 27.4 2.670 3.08  80 121 4       2
## 11 29.5 2.135 3.05  68  98 4       2
## 12 18.5 3.940 2.45 150 360 8       1
## 13 28.4 2.670 2.53  90 151 4       2
## 14 26.8 2.700 2.84 115 173 6       2
## 15 34.2 2.200 3.37  70 105 4       2
```

```
centers1 <- update(data1, centers)
centers1
```

```
##          B        C        D         E   F        G
## 1 17.70000 4.150000 2.590000 152.50000 355 8.000000
## 2 25.67692 2.811077 2.920769  95.07692 180 5.230769
```

We can find that some data records are devided to belong to cluster 1 and others are belong to cluster 2. Update the centers.

Second Round

Run the alogrithm for the second time.

```
data2 <- assign(data, centers)
data2
```

```
##       B     C    D   E   F G cluster
## 1  16.9 4.360 2.73 155 350 8       1
## 2  15.5 4.054 2.26 142 351 8       2
## 3  30.0 2.155 3.70  68  98 4       2
## 4  30.9 2.230 3.37  75 105 4       2
## 5  20.6 3.380 2.73 105 231 6       2
## 6  20.8 3.070 3.08  85 200 6       2
## 7  18.1 3.410 2.73 120 258 6       2
## 8  16.5 3.955 2.26 138 351 8       2
## 9  35.1 1.915 2.97  80  98 4       2
## 10 27.4 2.670 3.08  80 121 4       2
## 11 29.5 2.135 3.05  68  98 4       2
## 12 18.5 3.940 2.45 150 360 8       1
## 13 28.4 2.670 2.53  90 151 4       2
## 14 26.8 2.700 2.84 115 173 6       2
## 15 34.2 2.200 3.37  70 105 4       2
```

```r
centers2 <- update(data2, centers)
centers2
```

```
##          B        C        D         E   F        G
## 1 17.70000 4.150000 2.590000 152.50000 355 8.000000
## 2 25.67692 2.811077 2.920769  95.07692 180 5.230769
```

We can find that the centers are not changed. Thus all objects are divided into two clusters and the final clustering result is already got. The clustering result is:

```r
data2
```

```
##       B     C    D   E   F G cluster
## 1  16.9 4.360 2.73 155 350 8       1
## 2  15.5 4.054 2.26 142 351 8       2
## 3  30.0 2.155 3.70  68  98 4       2
## 4  30.9 2.230 3.37  75 105 4       2
## 5  20.6 3.380 2.73 105 231 6       2
## 6  20.8 3.070 3.08  85 200 6       2
## 7  18.1 3.410 2.73 120 258 6       2
## 8  16.5 3.955 2.26 138 351 8       2
## 9  35.1 1.915 2.97  80  98 4       2
## 10 27.4 2.670 3.08  80 121 4       2
## 11 29.5 2.135 3.05  68  98 4       2
## 12 18.5 3.940 2.45 150 360 8       1
## 13 28.4 2.670 2.53  90 151 4       2
## 14 26.8 2.700 2.84 115 173 6       2
## 15 34.2 2.200 3.37  70 105 4       2
```