K Means Clustering

Kumar Rahul

6/9/2017

Preparing Data

Read data from a specified location

```
car.data <- read.csv("/Users/Rahul/Documents/Datasets/Kmeans_Car data.csv",
header = TRUE,sep = ",",na.strings = c(""," ", "NA"))
```

Summary of the data on which model is built and Standardizing the variables

Information on the car metadata. column 3 to column 5 contains numeric data. In case of any other dataset change the column numbers accordingly

```
str(car.data)

## 'data.frame':    1008 obs. of  17 variables:
##  $ Brand           : Factor w/ 41 levels "Ashok Leyland",..: 38 38 38 38
38 38 38 38 38 38 ...
##  $ Car.Models      : Factor w/ 1008 levels "Ambassador CLASSIC 1500
DSL",..: 850 849 842 846 845 840 848 847 841 851 ...
##  $ Price..INR.     : int  141898 145000 150000 171489 191125 198000 198605
216238 223500 237831 ...
##  $ Mileage         : num  25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4 25.4
25.4 ...
##  $ Seating.Capacity: int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Vehicle.Type    : Factor w/ 9 levels "Convertible",..: 3 3 3 3 3 3 3 3
3 3 ...
##  $ Fuel.Type       : Factor w/ 5 levels "CNG","Diesel",..: 5 5 5 5 5 5 5 5
5 5 ...
##  $ Transmission    : Factor w/ 2 levels "Automatic","Manual": 2 2 2 2 2 2
2 2 2 2 ...
##  $ Parking.Sensor  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ Airbag          : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ Cruise.Control  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ Keyless.Entry   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 2 2
...
##  $ Alloy.wheels    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
```

```
##  $ ABS             : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ Climate.Control : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ Rear.AC.Vent    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1
...
##  $ Power.Steering  : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2
...

car.data <- na.omit(car.data) # listwise deletion of missing values
car.scaled <- scale(car.data[,3:5]) # standardize variables
```

Case to demonstrate dummy variable coding using *library(dummies)*

Using euclidean as distance measure for k means clustering

The Euclidian distance between any two observations within the cluster will be lesser than the observations between clusters. This is used to derive ideal number of clusters and quality of clusters.

Some of the metrics using this information is Calinski and Harabasz Index (CH Index).

CH(k) = [{B(k)/(k-1)}/{W(k)/(n-k)}]

Where CH(k) is the Calinski and Harabasz index with k-clusters (k > 1), B(k) and W(k) are the between and within clusters sum of squared variations with k clusters.The optimal K value is the one with maximum CH Index.

The other statistics which can be used is Silhouette width. Let a(i) be the average distance between an observation i and other points in the cluster to which observation i belongs. Let b(i) be the minimum average distance between observation i and observations in other clusters. Then the Silhouette statistic is defined by:

S(i) = [{b(i)-a(i)}/Max{a(i),b(i)}]

A higher value of S(i) indicates better clustering.

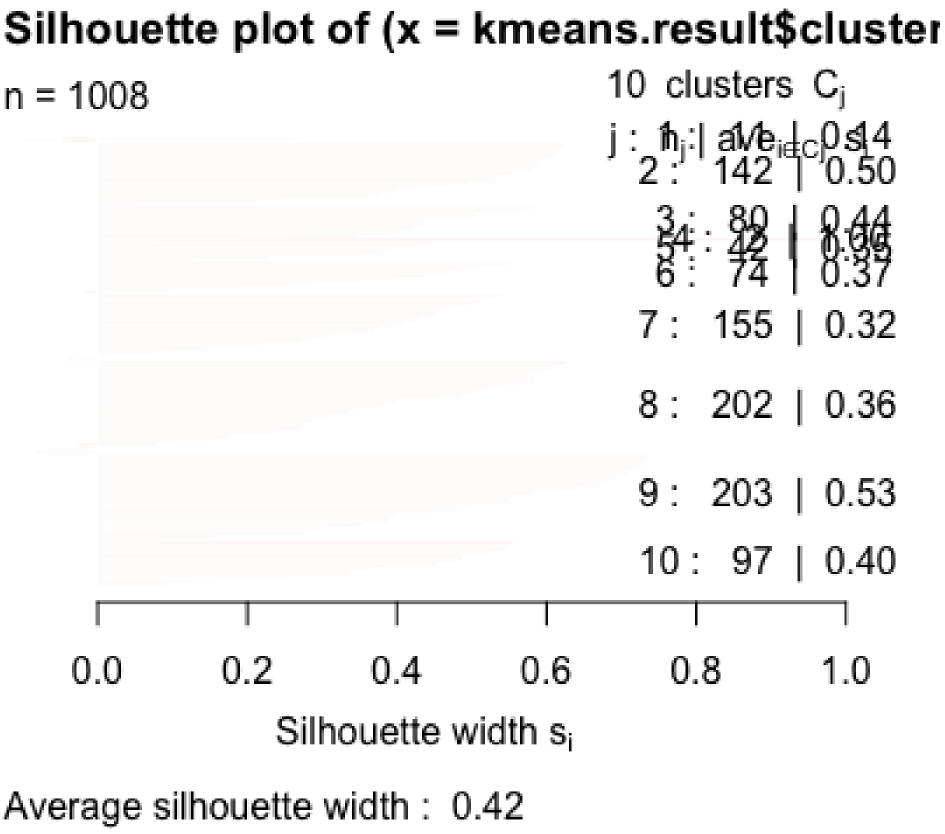
The loop is to demonstrate the use of making different number of clsuters. Here the loop is through 2 to 10. This implies that the code chunk is making two clsuters to ten clusters.

```
temp.car <- car.data[,1:5]
distance.metric <-  dist(car.scaled, method = "euclidean") # distance matrix
for (i in 2:10) {
  kmeans.result <- kmeans(car.scaled, centers=i)

  #silhouette statistics as a measure of quality of clusters
  sil.width  <- silhouette(kmeans.result$cluster, distance.metric)
  summary.sil.width <- summary(sil.width)

  #CH Index as a measure of quality of cluter
  temp1 <- kmeans.result$betweenss/(i-1)
  temp2 <- kmeans.result$tot.withinss/(1008-i)
  ch <-  temp1/temp2

  #Append the silhouette stats and CH index along with cluster number.
  cluster.member <-
cbind(kmeans.result$cluster,sil.width[,3],summary.sil.width$avg.width,ch)
  colnames(cluster.member)[1] <- paste("ClusterOf",i)
  colnames(cluster.member)[2] <- paste("SilWidth-",i)
  colnames(cluster.member)[3] <- paste("AvgClusterWidth-",i)
  colnames(cluster.member)[4] <- paste("CHIndex-",i)
  temp.car <- data.frame(temp.car,cluster.member)
  #print(temp.car)
}
```

Plot the result: Silhouette plot to visulaize the clusters

kmeans.results from above code chunk is being used. Since it is a loop through in the above code. K = 10 is being used in below code chunk.

```
distance.metric <-  dist(car.scaled, method = "euclidean") # distance matrix
#d <- daisy(car.scaled, metric = c("euclidean")) #can use daisy as well
sil.width   <- silhouette(kmeans.result$cluster, distance.metric)
plot(sil.width, col = "red")
```



Write the result to csv

```r
write.csv(temp.car, "kmeans-analysis.csv") #export to excel
```