Data Analytics Project (Clustering Diabetes)

Amirehsan Davoodi

May 10, 2018

Dataset:

The data lists various attributes of people diagnosed with diabetes from the year 1999 to 2008 in 130 US hospitals.

- Load the dataset

```
data(iris)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

- Remove the values from the **species** columns

```
iris_species_unk = iris
iris_species_unk$Species = NULL
```

K-Means

- The `nstart` allows to run different random starting assigments and to select the one with the lowest within cluster variation
- Ensure reproducibility by setting the seed
- Assume that $K = 3$

```
# kmeans(x, centers, ...)
set.seed(20)
km_clusters = kmeans(iris_species_unk[], centers = 3, nstart = 20)
str(km_clusters)
```

```
## List of 9
##  $ cluster     : int [1:150] 3 3 3 3 3 3 3 3 3 3 ...
##  $ centers     : num [1:3, 1:4] 6.85 5.9 5.01 3.07 2.75 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:4] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width"
##  $ totss       : num 681
##  $ withinss    : num [1:3] 23.9 39.8 15.2
##  $ tot.withinss: num 78.9
##  $ betweenss   : num 603
##  $ size        : int [1:3] 38 62 50
##  $ iter        : int 2
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

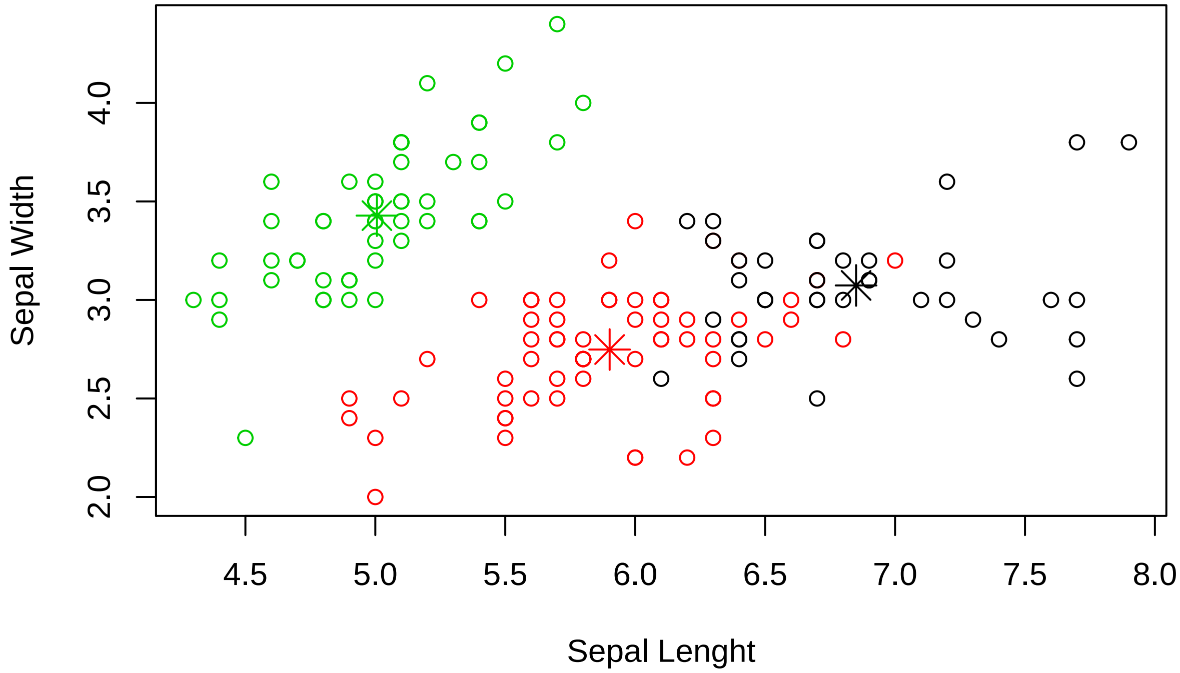
- Compare the clusters with the species

```r
table(km_clusters$cluster, iris$Species)
```

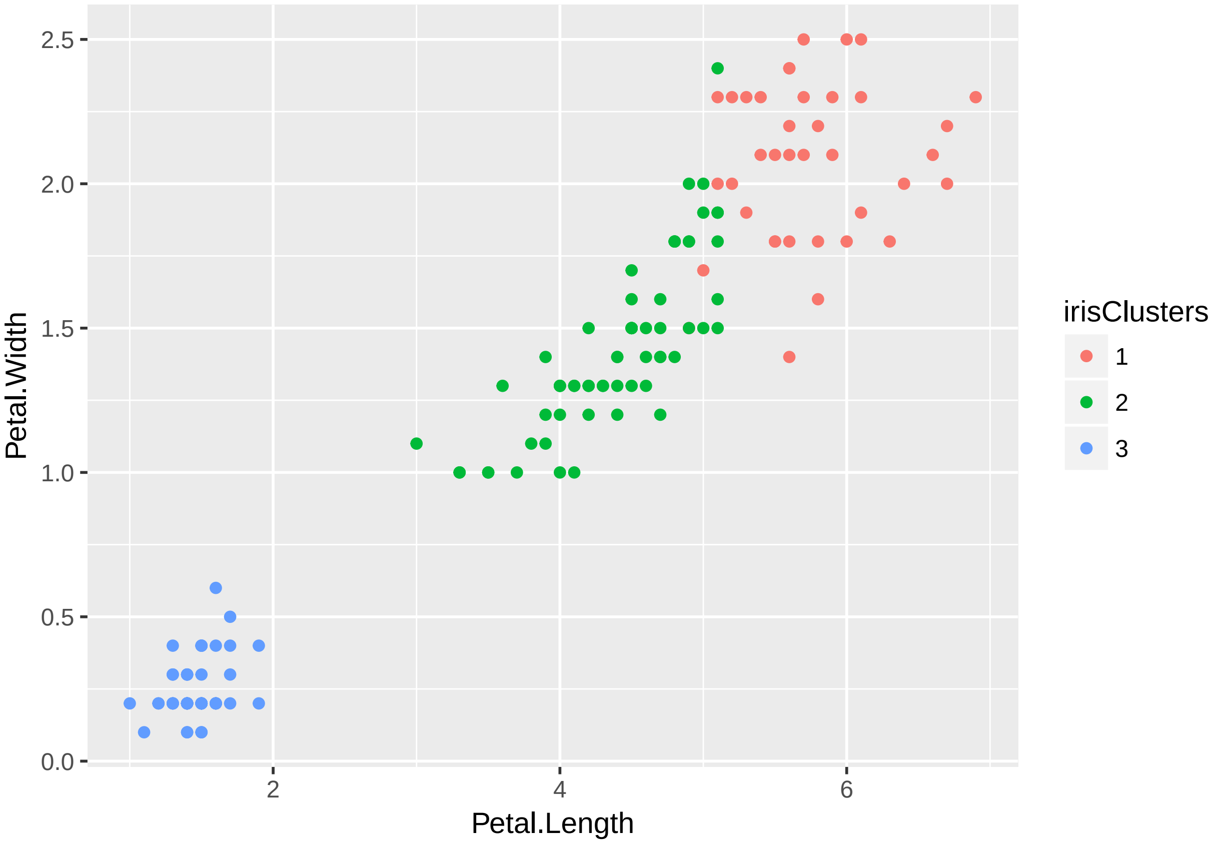
```
##
##     setosa versicolor virginica
##   1      0          2        36
##   2      0         48        14
##   3     50          0         0
```

- Plot data samples in clusters

```r
plot(iris_species_unk$Sepal.Length, iris_species_unk$Sepal.Width, col=km_clusters$cluster, xlab = 'Sepal
points(km_clusters$centers[,c('Sepal.Length', 'Sepal.Width')], col=1:3, pch=8, cex=2)
```



```r
library(ggplot2)
irisClusters = as.factor(km_clusters$cluster)
ggplot(iris_species_unk, aes(Petal.Length, Petal.Width, color = irisClusters)) + geom_point()
```



Optimal K computation

Model selection criteria:

- **AIC** (Akaike Information Criterion)
- **BIC** (Bayesian Information Criterion)

```
aic_bic = function(fit){
  # Number of features #
  m = ncol(fit$centers)
  # Number of observations #
  n = length(fit$cluster)
  # Number of clusters, i.e. k #
  k = nrow(fit$centers)
  # Total within-cluster sum of squares
  D = fit$tot.withinss
  return(c(D + 2*m*k, D + log(n)*m*k))
}
```

- Which are the **AIC** and **BIC** values?

```
values = aic_bic(km_clusters)
names(values) = c('AIC', 'BIC')
print(values)
```

```
##      AIC      BIC
## 102.8514 138.9791
```

- Pick the model with the lowest **BIC** or **AIC**

- Check values of **K** between 3 to 40

```
#cat("K", "\t", "AIC", "\t\t", "BIC", "\n")
lowest_bic = lowest_aic = 1000
best_k_bic = best_k_aic = 0
for (k in 3:40) {
  aic_bic_k = aic_bic(kmeans(iris_species_unk, k))
  current_aic = aic_bic_k[1]
  current_bic = aic_bic_k[2]
  if (current_aic < lowest_aic) {
    lowest_aic = current_aic
    best_k_aic = k
  }
  if (current_bic < lowest_bic) {
    lowest_bic = current_bic
    best_k_bic = k
  }
  # cat(k, '\t', bic_aic_k[1], '\t', bic_aic_k[2], '\n')
}
```

```
cat('Best K according to AIC: ', best_k_aic, '-- BIC: ', lowest_aic, '\n')
```

```
## Best K according to AIC:  5 -- BIC:  90.13655
```

```
cat('Best K according to BIC: ', best_k_bic, '-- BIC: ', lowest_bic, '\n')
```

```
## Best K according to BIC:  3 -- BIC:  138.9791
```

Activity:

Apply the same analysis as before to the following dataset:

- `install.packages('rattle.data')`
- `data("wine", package = 'rattle.data')`

Hierarchical Clustering

- Take a sample from the IRIS dataset:

```
idx = sample(1:dim(iris)[1], 40)
iris_sample = iris[idx,]
iris_sample$Species = NULL
```

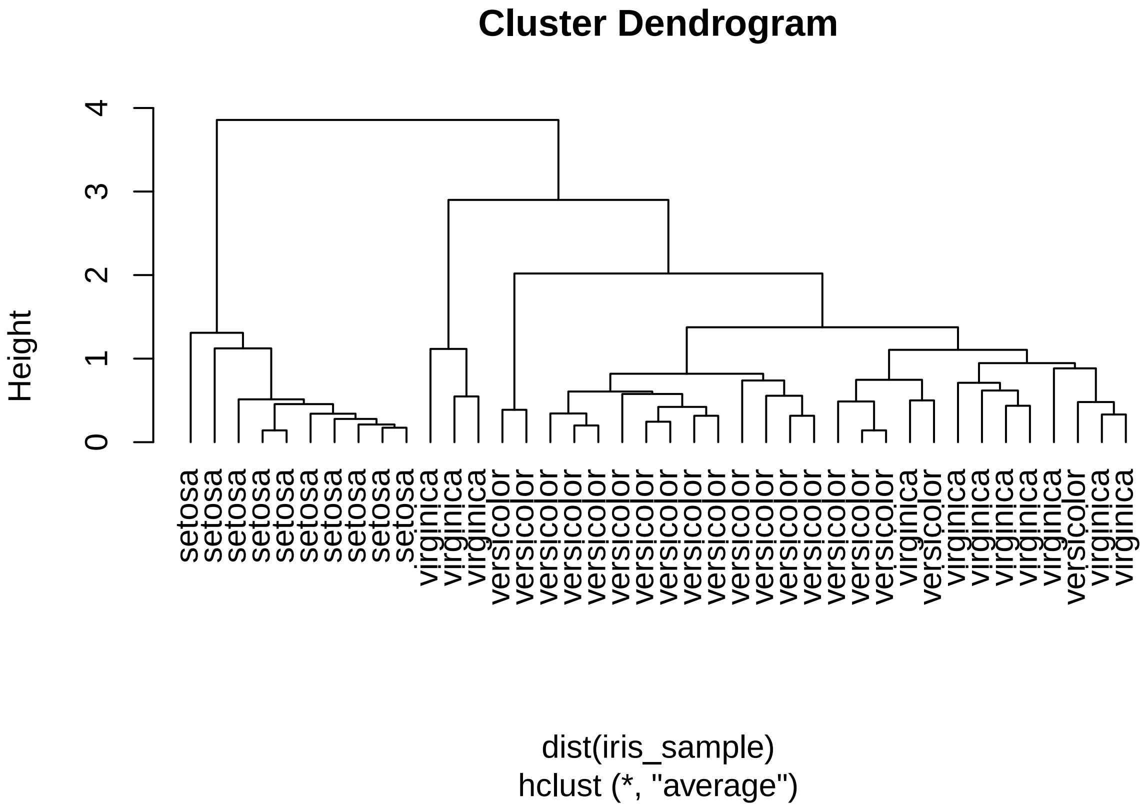
- Create the clusters

```
hc_clusters = hclust(dist(iris_sample), method="ave")
str(hc_clusters)
```

```
## List of 7
##  $ merge      : int [1:39, 1:2] -6 -21 -18 -3 -40 -16 -13 -2 -5 -9 ...
##  $ height     : num [1:39] 0.141 0.141 0.173 0.2 0.212 ...
##  $ order      : int [1:40] 22 15 12 6 34 33 13 40 18 24 ...
##  $ labels     : chr [1:40] "112" "56" "79" "134" ...
##  $ method     : chr "average"
##  $ call       : language hclust(d = dist(iris_sample), method = "ave")
##  $ dist.method: chr "euclidean"
##  - attr(*, "class")= chr "hclust"
```

- Plot the result

```
plot(hc_clusters, hang = -1, labels=iris$Species[idx])
```


Cluster Dendrogram
Height
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
setosa
virginica
virginica
virginica
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
versicolor
virginica
versicolor
virginica
virginica
virginica
virginica
versicolor
virginica
virginica
dist(iris_sample)
hclust (*, "average")