19 - Cluster Analysis - k Means Method

Junvie Pailden

SIUE, F2017, Stat 589

November 15, 2017

K-Means Method (Nonhierarchical Clustering)

1. Partition the items into K initial clusters.
2. Proceed through the list of items, assigning an item to the cluster whose centroid (mean) is nearest (usually used Euclidean distance).
   - Recalculate the centroid for the cluster receiving the new item and for the cluster losing the item.
3. Repeat Step 2 until no more reassignments take place.
   - Rather than starting with a parition of all items into K preliminary groups in Step 1, we could specify K initial centroids (seed points) and then proceed to Step 2.
   - Final assignment is dependent on the initial partition.

Example 12.11 Clustering using the K-means method

Suppose we measure two variables X_1 and X_2 for each of foour items, A, B, C, and D.

Observations		
Item	x_1	x_2
A	5	3
B	-1	1
C	1	-2
D	-3	-2

- Divide the items into $K = 2$ clusters such that the items within each cluster are closer to one another thatn they are to the items in different clusters.

Example 12.11 (cont)

- At Step 2, we compute the Euclidean distance of each item from the groupcentroids and reassign each item to the nearest group.
- If an item is moved from the initial configuration, the cluster centroids (means) must be updated before proceeding.
- The ith coordinate, $i = 1, 2, \ldots, p$, of the centroid is easily updates using the formulas:

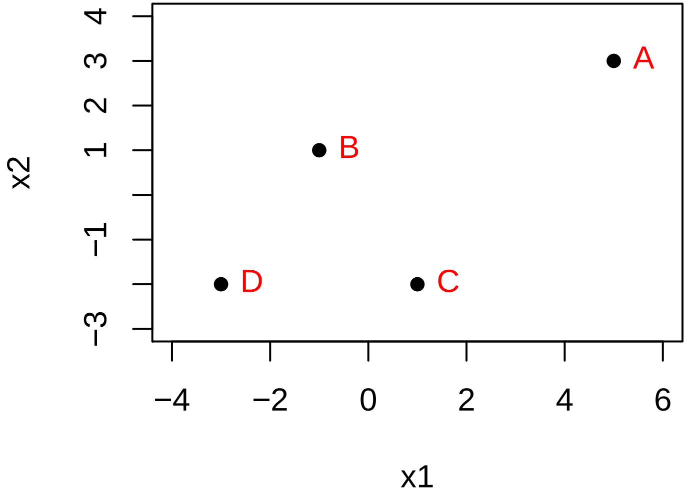
$$\bar{x}_{i,new} = \begin{cases} \frac{n\bar{x}_i + x_{ji}}{n+1} & \text{if the jth item is added to a group} \\ \frac{n\bar{x}_i - x_{ji}}{n-1} & \text{if the jth item is removed to a group} \end{cases}$$

where n is the number of items in the old group with centroid $\bar{x}' = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p)$.

Example 12.11 I

```
x1 <- c(5, -1, 1, -3)
x2 <- c(3, 1, -2, -2)
X0 <- cbind(x1, x2)
plot(X0, pch =16, xlim = c(-4, 6), ylim = c(-3, 4))
text(X0, c("A", "B", "C", "D"), pos = 4, col = "red")
```

Example 12.11 II



Example 12.11 (Kmeans function) I

```
(fit0 <- kmeans(X0, centers = 2))
```

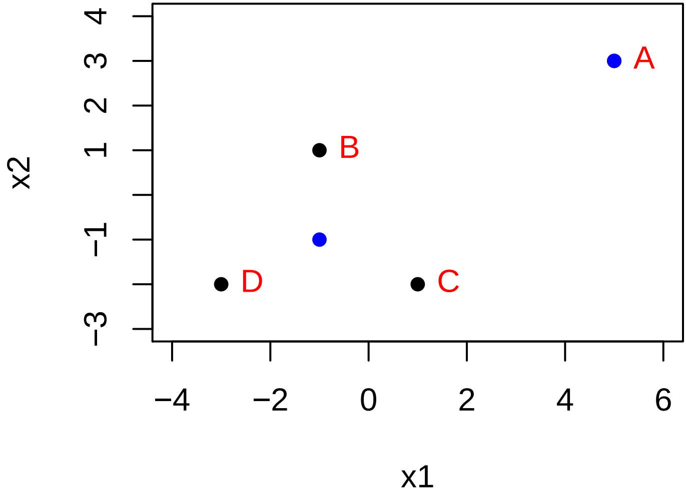
```
# K-means clustering with 2 clusters of sizes 3, 1
#
# Cluster means:
#   x1 x2
# 1 -1 -1
# 2  5  3
#
# Clustering vector:
# [1] 2 1 1 1
#
# Within cluster sum of squares by cluster:
# [1] 14  0
#  (between_SS / total_SS =  73.6 %)
```

Example 12.11 (Kmeans function) II

```
#
# Available components:
#
# [1] "cluster"      "centers"      "totss"        "withins
# [5] "tot.withinss" "betweenss"    "size"         "iter"
# [9] "ifault"
```

```
plot(X0, pch =16, xlim = c(-4, 6), ylim = c(-3, 4))
text(X0, c("A", "B", "C", "D"), pos = 4, col = "red")
points(fit0$centers, col = "blue", pch = 16)
```

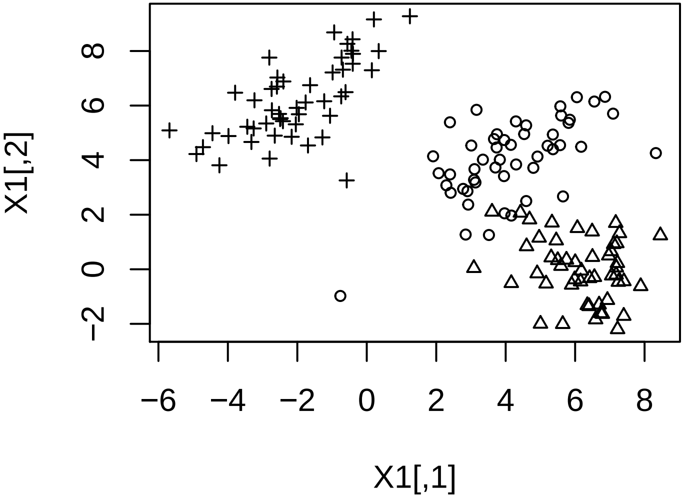
Example 12.11 (Kmeans function) III



Syntethic Data Example I

```r
library(MASS)
# generate a multivariate normal data
Sigma1 <- matrix(c(2.25,1.5,1.5,2.25), nrow=2)
set.seed(17)
group1 <- mvrnorm(50, mu = c(4, 4), Sigma = Sigma1)
group2 <- mvrnorm(50, mu = c(6, 0), Sigma = diag(2))
group3 <- mvrnorm(50, mu = c(-2, 6), Sigma = Sigma1)
X1 <- rbind(group1, group2, group3)
class1 <- c(rep(1, 50), rep(2, 50), rep(3, 50))
plot(X1, pch = class1, cex = 0.7)
```

Syntethic Data Example II



Random starts can give different predicted cluster I

```r
set.seed(17)
fit11 <- kmeans(X1, centers = 3)
table(fit11$cluster)
```

```
#
#  1  2  3
# 53 50 47
```

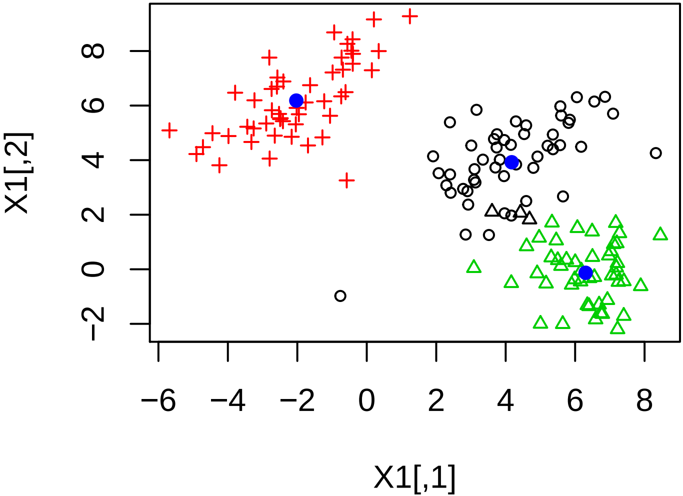
```r
fit11$centers # cluster centers
```

```
#    [,1]   [,2]
# 1  4.17  3.923
# 2 -2.03  6.183
# 3  6.30 -0.129
```

Random starts can give different predicted cluster II

```r
plot(X1, pch = class1, col = fit11$cluster, cex = 0.7)
points(fit11$centers, col = "blue", pch = 16)
```

Random starts can give different predicted cluster III



Different Seed, different predicted cluster I

```r
set.seed(18)
fit12 <- kmeans(X1, centers = 3)
table(fit12$cluster)
```

```
#
#  1  2  3
# 96 20 34
```

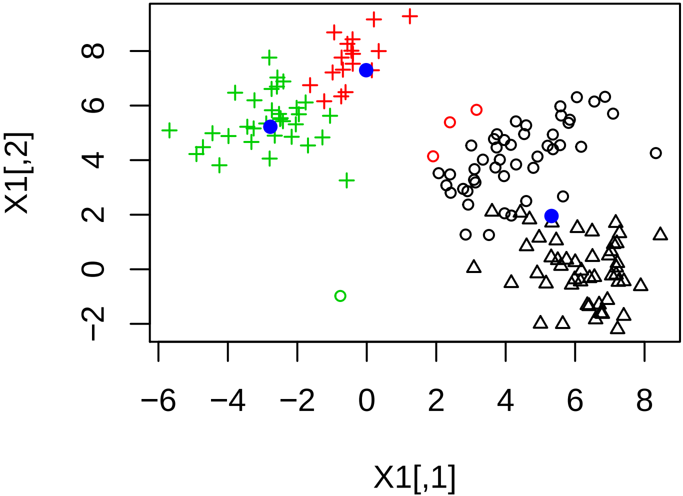
```r
fit12$centers
```

```
#      [,1] [,2]
# 1  5.3198 1.95
# 2 -0.0175 7.30
# 3 -2.7773 5.22
```

Different Seed, different predicted cluster II

```
plot(X1, pch = class1, col = fit12$cluster, cex = 0.7)
points(fit12$centers, col = "blue", pch = 16)
```

Different Seed, different predicted cluster III



Fixed on random start issue? I

- The `nstart` argument tells `kmeans` to try many random starts and keep the best.
- With 20 or 25 random starts, you'll generally find the overall best solution unless your sample size is really big.

```
fit21 <- kmeans(X1, centers = 3, nstart = 25)
fit22 <- kmeans(X1, centers = 3, nstart = 25)
# heirarchical clustering method using Ward Method
fit2.hclust <- hclust(dist(X1), method = "ward.D")
# similar kmeans predicted cluster
table(kmeans1 = fit21$cluster, kmeans2 = fit22$cluster)
```

Fixed on random start issue? II

```
#          kmeans2
# kmeans1   1  2  3
#         1  0 50  0
#         2 53  0  0
#         3  0  0 47
```

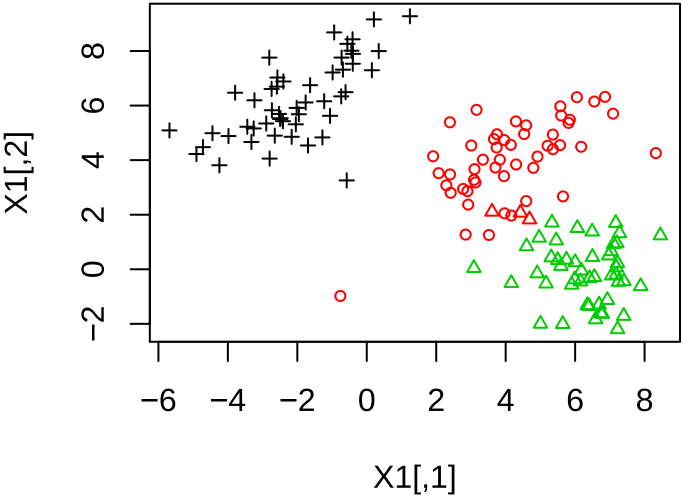
```
# kmeans and Ward Heirarchical clustering method
table(kmeans1 = fit21$cluster,
      hclust = cutree(fit2.hclust, k = 3))
```

```
#          hclust
# kmeans1   1  2  3
#         1  0  0 50
#         2 53  0  0
#         3  5 42  0
```

Fixed on random start issue? III

```r
# lets see kmeans clusters output
plot(X1, pch = class1, col = fit21$cluster, cex = 0.7)
```

Fixed on random start issue? IV



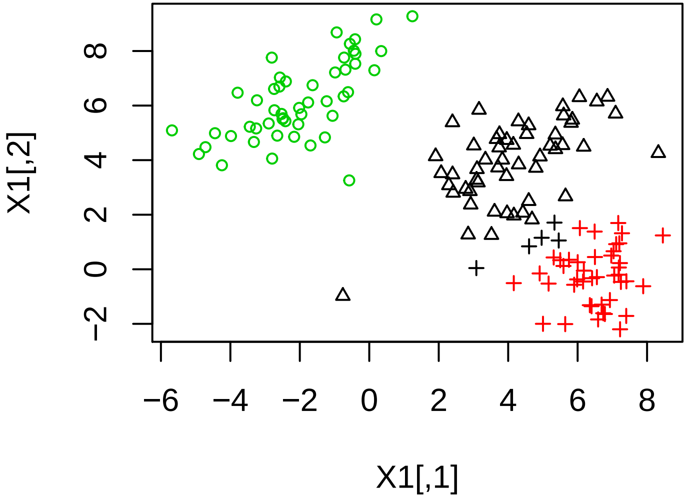
Comparison between heirarchical clustering and kmeans I

```r
# heirarchical clustering method using Ward Method
fit2.hclust <- hclust(dist(X1), method = "ward.D")
# predicted clusters are almost the same
table(kmeans1 = fit21$cluster,
      hclust = cutree(fit2.hclust, k = 3))
```

```
#          hclust
# kmeans1   1  2  3
#        1  0  0 50
#        2 53  0  0
#        3  5 42  0
```

```r
# lets see kmeans and hclust clusters output
plot(X1, pch = fit21$cluster,
      col = cutree(fit2.hclust, k = 3), cex = 0.7)
```

Comparison between heirarchical clustering and kmeans II



Example 12.12 K-means clustering of public utilities I

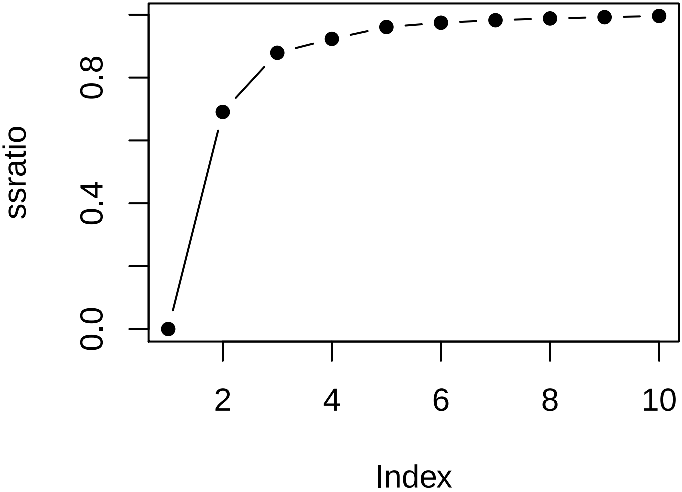
- The K-means algorithm for several choices of K was run.
- The choise of K depends upon the subject-matter knowedgge as well choosing K so as to maximize the between-cluster variability relative to the within-cluster variability, such as

$$SS_{ratio} = \frac{|\boldsymbol{W}|}{|\boldsymbol{B} + \boldsymbol{W}|} \quad \text{and} \quad tr(\boldsymbol{W}^{-\mathbf{1}}\boldsymbol{B})$$

Example 12.12 K-means clustering of public utilities II

```
utility <- as.matrix(read.table("T12-5.DAT")[,1:8])
rownames(utility) <- read.table("T12-5.DAT")[,9]
ssratio <- rep(NA, 10) # create storage vector
for(k in 1:length(ssratio)) {
    # Tries numerous random starts
    fit = kmeans(utility, k, nstart = 25)
    ssratio[k] <- fit$betweenss/fit$totss
}
# Levels out after k = 3
plot(ssratio, type = "b", pch = 16)
```

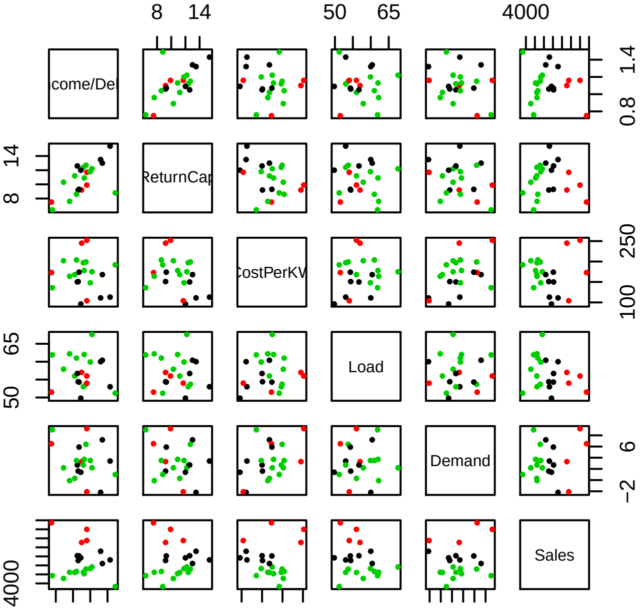
Example 12.12 K-means clustering of public utilities III



K-means clustering of public utilities

```
colnames(utility) <- c("Income/Debt", "ReturnCap",
                       "CostPerKW", "Load", "Demand",
                       "Sales", "PercNuc", "FuelC")
# use 3 clusters
util.kmeans <- kmeans(utility, 3, nstart = 25)
# compare with agglomerative clustering
```

```r
pairs(utility[, 1:6], col = util.kmeans$cluster,
      pch = 16, cex = 0.6)
```



come/Del
ReturnCap
CostPerKV
Load
Demand
Sales
8
14
50
65
4000
1.4
0.8
250
100
6
−2

Swiss Canton Data I

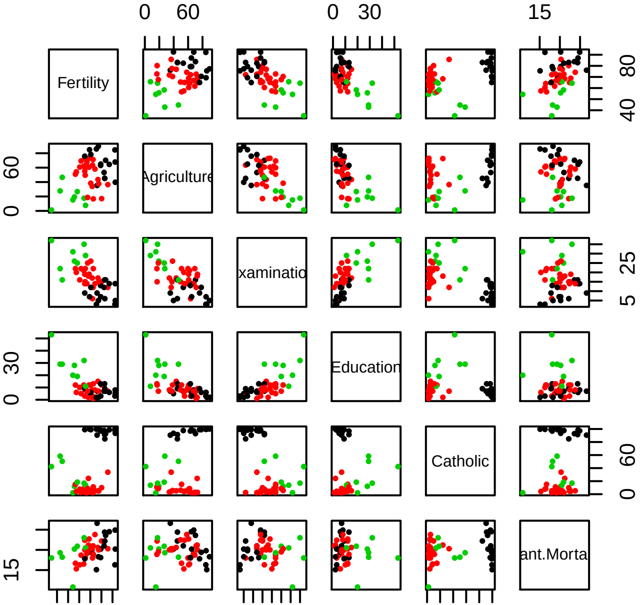
Switzerland, in 1888, was entering a period known as the demographic transition; i.e., its fertility was beginning to fall from the high level typical of underdeveloped countries. The data collected are for 47 French-speaking “provinces” at about 1888.

```
names(swiss)
```

```
# [1] "Fertility"        "Agriculture"      "Examination"
# [4] "Education"        "Catholic"         "Infant.Mortali
```

```
# use 3 clusters
swiss.kmeans <- kmeans(scale(swiss), 3, nstart=25)
```

```r
pairs(swiss, col = swiss.kmeans$cluster,
      pch = 16, cex = 0.6)
```


Fertility
Agriculture
xaminatio
Education
Catholic
ant.Morta
0
60
0
30
15
80
40
60
0
25
5
30
0
60
0
15