Ruoxi_Liu_Assignment

Ruoxi Liu

3/28/2019

Step 1

Simulate high-dimensional data (p=1000) with three groups of observations where the number of observations is n=100.

```
library(clues)
library(ggplot2)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 3.3.2
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(purrr)


n_rows = 1000
n_cols = 100

k=3
x_mus = c(0,5,5)
x_sds = c(1,0.1,1)
y_mus = c(5,5,0)
y_sds = c(1,0.1,1)
prop1 = c(0.3,0.5,0.2)

comp1 <- sample(seq_len(k), prob=prop1, size=n_cols, replace=TRUE)
samples1 <- cbind(rnorm(n=n_cols, mean=x_mus[comp1],sd=x_sds[comp1]),
                  rnorm(n=n_cols, mean=y_mus[comp1],sd=y_sds[comp1]))

proj <- matrix(rnorm(n_rows* n_cols), nrow=n_rows, ncol=2)
A1 <- samples1 %*% t(proj)
A1 <- A1 + rnorm(n_rows*n_cols)
```

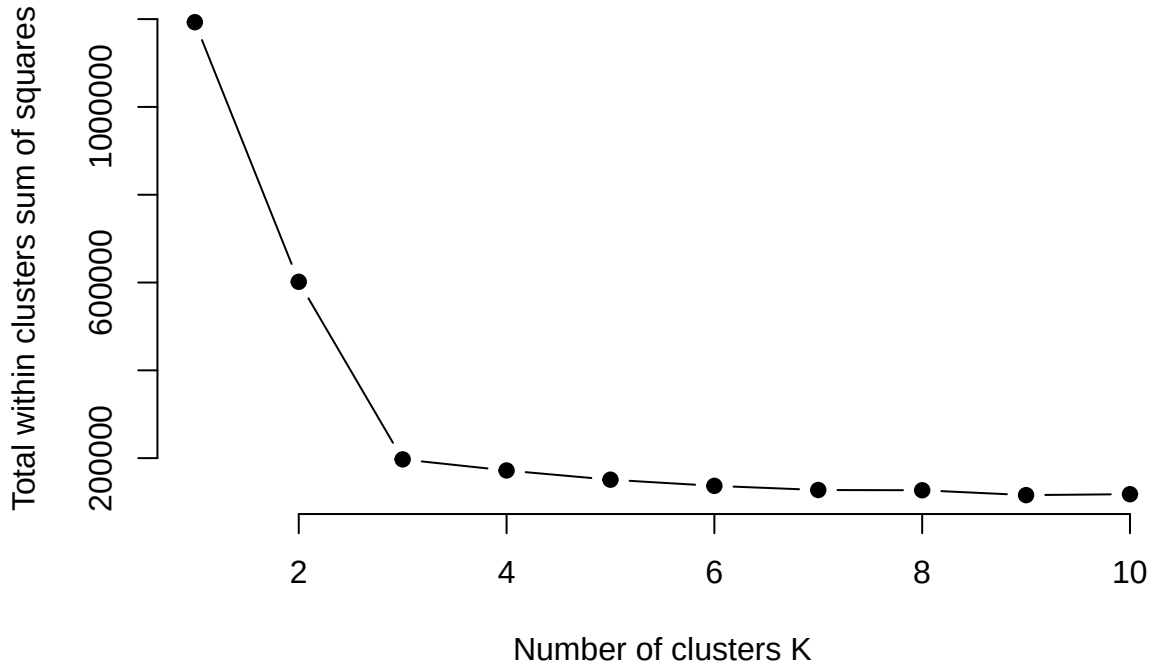
Step 2

Perform k-means. To decide what k to choose, I performed k-means 10 times, using the same dataset A1, with k equals to 1 to 10 respectively. Then I plotted the total within clusters sum of squares.

```
set.seed(123)
GetWss <- function(k) {
  kmeans(A1, k, nstart = 10 )$tot.withinss
}
k.values <- 1:10
wss_values <- map_dbl(k.values, GetWss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
```

```
      xlab="Number of clusters K",
      ylab="Total within clusters sum of squares")
```



From the graph, we could see the within clusters sum of squares (wss) decreases significantly when move from k = 2 to k = 3. And the decrease is not big after k = 3. Since the wss measures the deviations between each data point and the cluster center, a small wss suggests that the clustering algorithm successfully places the similar data points into a cluster. Thus, to achieve a good performance of the k means clustering, I decide to use k = 3, i.e., there are 3 clusters in the data.

Step 3

Perform the k means 100 times with the randomly generated data (100 different datasets), with k = 3. Compute adjusted rand index and within clusters sum of squares to assess the accuracy.

```
ari <- c()
twss <- c()

for(i in 1:100) {
  n_rows = 1000
  n_cols = 100

  k=3
  x_mus = c(0,5,5)
  x_sds = c(1,0.1,1)
  y_mus = c(5,5,0)
  y_sds = c(1,0.1,1)
  prop1 = c(0.3,0.5,0.2)

  comp1 <- sample(seq_len(k), prob=prop1, size=n_cols, replace=TRUE)
  samples1 <- cbind(rnorm(n=n_cols, mean=x_mus[comp1],sd=x_sds[comp1]),
                    rnorm(n=n_cols, mean=y_mus[comp1],sd=y_sds[comp1]))
```

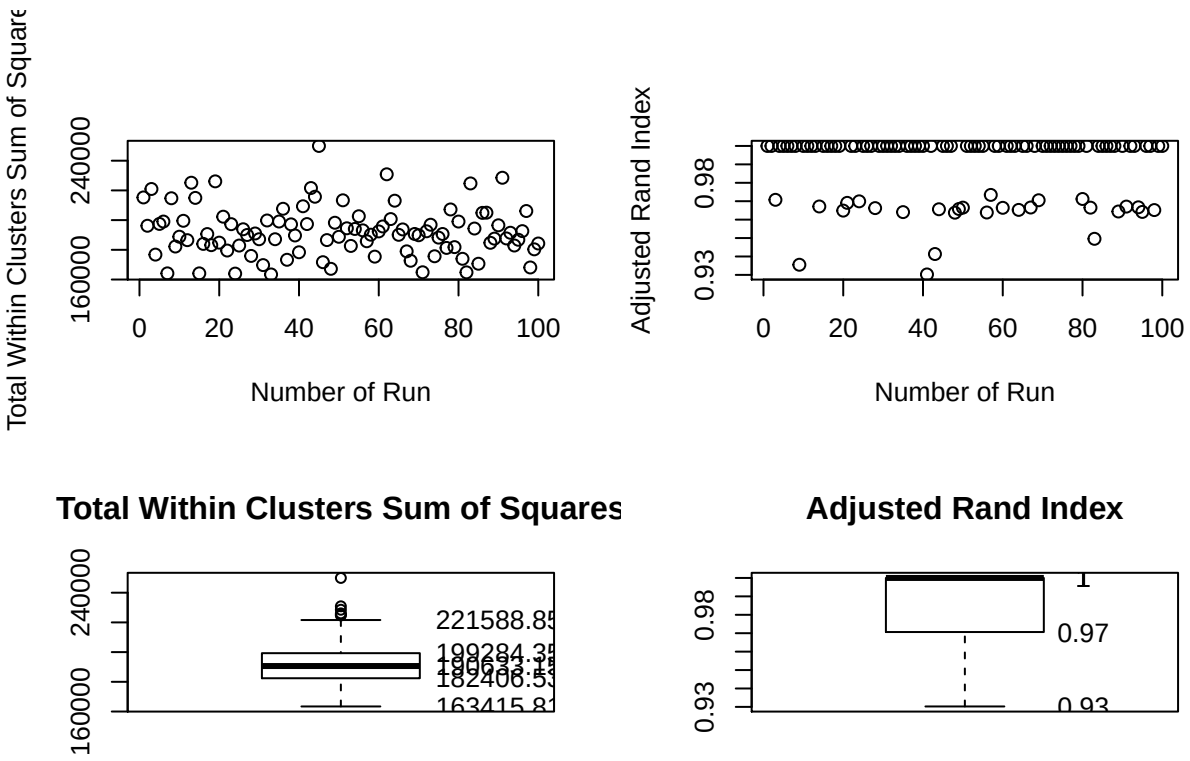
```
  proj <- matrix(rnorm(n_rows* n_cols), nrow=n_rows, ncol=2)
  A1 <- samples1 %*% t(proj)
  A1 <- A1 + rnorm(n_rows*n_cols)

  k2 <- kmeans(A1, centers = 3, nstart=25)
  twss[i] <- k2$tot.withinss
  ari[i] <- adjustedRand(k2$cluster, comp1)[2]
}
```

Step 4

Visualize the adjusted rand index and the within clusters sum of squares recorded in the 100 runs.

```
xaxis <- 1:100
par(mfrow=c(2,2))
plot(xaxis, twss, xlab = "Number of Run", ylab = "Total Within Clusters Sum of Squares")
plot(xaxis, ari, xlab = "Number of Run", ylab = "Adjusted Rand Index")
boxplot(twss, main="Total Within Clusters Sum of Squares")
text(y = round(boxplot.stats(twss)$stats,2), labels = round(boxplot.stats(twss)$stats,2), x = 1.4)
boxplot(ari,main="Adjusted Rand Index")
text(y = round(boxplot.stats(ari)$stats,2), labels = round(boxplot.stats(ari)$stats,2), x = 1.3)
```




```
sum(ari == 1)
```

```
## [1] 72
```

From the adjusted rand index and the within clusters sum of squares, we could see that the k means clustering algorithm does a pretty good job in clustering the data points. The adjusted rand index (ARI) measures the similarity between the clusters assigned by k means and the real clusters. The range of ARI is from 0.93 to 1, where 1 means the algorithm put every data point into a correct cluster. Specifically, among the

100 runs, there are 72 times that the k means algorithm achieves 100% accuracy. Next, looking at the total within clusters sum of squares, the average total wss is 190633.15, which is pretty good, considering the size of dimensions of the original dataset. Thus, I believe the k means algorithm is a good model to cluster this dataset.