Coding Assignment

Xiaoyu Qiao

3/26/2019

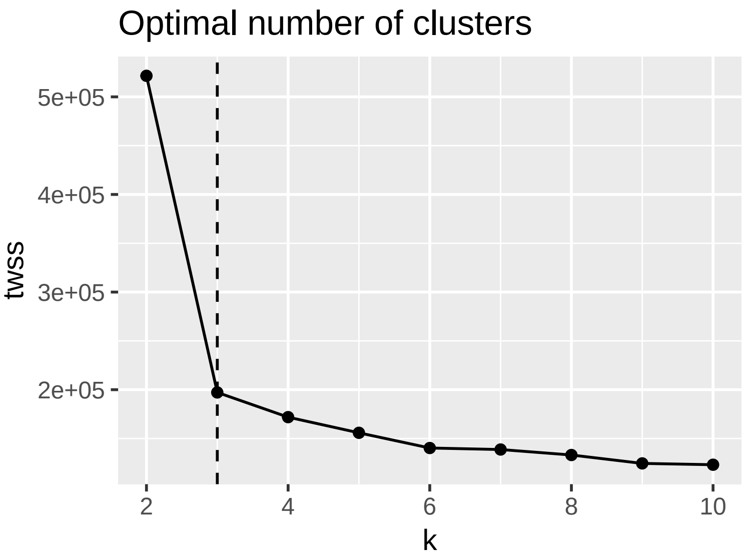
```
library(mclust)
library(cluster)
library(ggplot2)
```

We firstly write a function getA1 for simulating high-dimensional data (p=1000) with three groups of observations where the number of observations is n=100:

```
getA1 <- function(){
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)
return (list("data" = A1, "labels" = comp1))
}
```

We firstly take a look at a single run, and find out the optimal number of clustering; we plot total within groups sum of squares against values of k, we pick k to be the elbow point, which corresponding to $k = 3$.

```
set.seed(1)

result = getA1()
A1=result$data

#function for calculating total within groups sum of squares
twss <- function(fit){
return(fit$tot.withinss)
}

result = data.frame(k=c(2:10),twss=sapply(2:10,function(k){twss(kmeans(A1, k,nstart = 25))}))
ggplot(data=result, aes(x=k, y=twss)) + geom_line()+geom_point()+ geom_vline(xintercept = 3,linetype =
```



```
k.opt=3
```

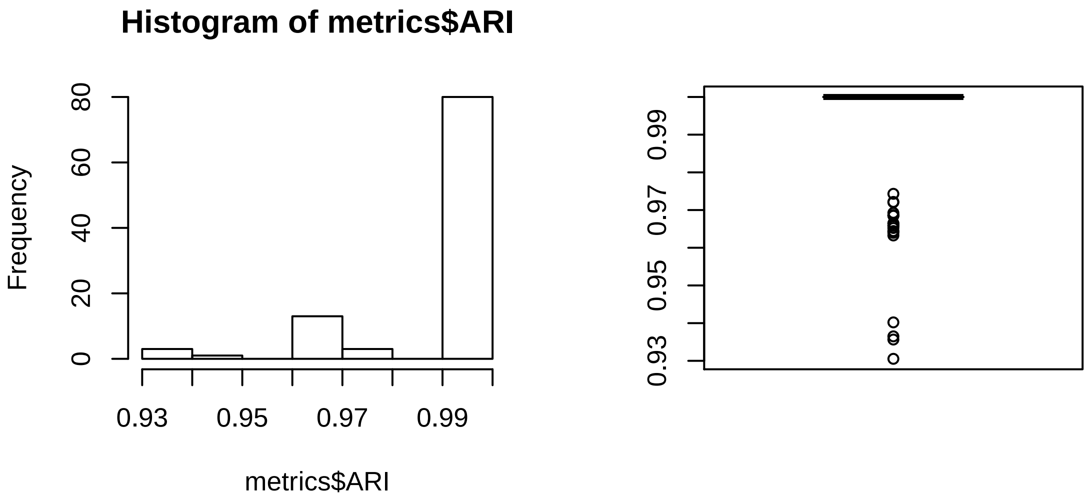
We generate simulated high-dimensional data and perform K-means 100 times; and we calculate the adjusted rand index and the total within clusters sum of squares for each run:

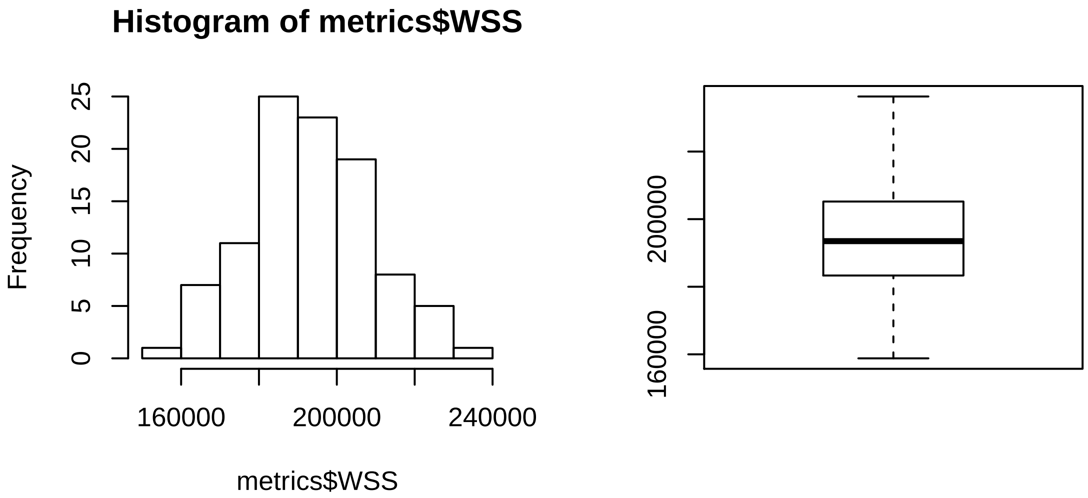
```
metrics <- data.frame(ARI=numeric(0),WSS=numeric(0))

for (i in 1:100) {
result = getA1()
A1 = result$data
lbs = result$labels
KM = kmeans(A1, k.opt,nstart = 25)
clusters <- KM$cluster
new <- data.frame(adjustedRandIndex(clusters, lbs), twss(KM))
names(new)<-c("ARI","WSS")
metrics <- rbind(metrics,new)
}
metrics
```

```
##        ARI    WSS
## 1   1.0000 195850
## 2   1.0000 168021
## 3   0.9356 196977
## 4   1.0000 193370
## 5   1.0000 183157
## 6   1.0000 175918
## 7   1.0000 171186
## 8   1.0000 186844
## 9   1.0000 184879
## 10  1.0000 181435
## 11  1.0000 172146
## 12  1.0000 191679
## 13  1.0000 185770
## 14  1.0000 162601
## 15  1.0000 196046
## 16  0.9722 206972
## 17  0.9644 213957
## 18  1.0000 210627
## 19  1.0000 174344
## 20  1.0000 236300
## 21  0.9365 219279
## 22  1.0000 178922
## 23  1.0000 199970
## 24  1.0000 184080
## 25  1.0000 165469
## 26  1.0000 173994
## 27  1.0000 192761
## 28  1.0000 200808
## 29  1.0000 185874
## 30  1.0000 228024
## 31  1.0000 209945
## 32  1.0000 212046
## 33  0.9686 172977
## 34  0.9658 204462
## 35  1.0000 189869
## 36  1.0000 183481
## 37  1.0000 212827
## 38  1.0000 182801
## 39  1.0000 192912
## 40  1.0000 169618
## 41  0.9685 174324
## 42  1.0000 208254
## 43  0.9402 185029
## 44  1.0000 199699
## 45  1.0000 193614
## 46  0.9652 194784
## 47  1.0000 200334
## 48  1.0000 210419
## 49  0.9664 205230
## 50  1.0000 197194
## 51  0.9644 186062
## 52  1.0000 183654
## 53  1.0000 165188
## 54  0.9659 222213
## 55  1.0000 188083
## 56  1.0000 182992
## 57  1.0000 207603
## 58  1.0000 194339
## 59  0.9665 209267
## 60  0.9743 208061
## 61  1.0000 192761
## 62  1.0000 186339
## 63  1.0000 205345
## 64  1.0000 184401
## 65  1.0000 205249
## 66  1.0000 193820
## 67  1.0000 185795
## 68  1.0000 197389
## 69  1.0000 202249
## 70  1.0000 205306
## 71  1.0000 197607
## 72  1.0000 185899
## 73  1.0000 178066
## 74  1.0000 190200
## 75  1.0000 195433
## 76  1.0000 179916
## 77  0.9305 207111
## 78  1.0000 196342
## 79  1.0000 180300
## 80  1.0000 200982
## 81  1.0000 183801
## 82  0.9721 180053
## 83  1.0000 204807
## 84  0.9640 197556
## 85  1.0000 205202
## 86  1.0000 210403
## 87  0.9633 227681
## 88  1.0000 194895
## 89  0.9693 227222
## 90  1.0000 187314
## 91  1.0000 187858
## 92  1.0000 184723
## 93  1.0000 217130
## 94  1.0000 167672
## 95  0.9687 158808
## 96  1.0000 193799
## 97  1.0000 205198
## 98  1.0000 166794
## 99  1.0000 228617
## 100 1.0000 172329
```

Now we use boxplot and histogram to view the result of adjusted rand index and the total within clusters sum of squares:

```r
par(mfrow=c(2,2))
hist(metrics$ARI)
boxplot(metrics$ARI)
hist(metrics$WSS)
boxplot(metrics$WSS)
```





By the result of adjusted rand index, we know our K-means model has great accuracy.