Chapter 2:K-means Algorithms and Labs

Kevin Kam Fung Yuen
PhD, Senior Lecturer, School of Business, Singapore University of Social Sciences
kfyuen@suss.edu.sg, kevinkf.yuen@gmail.com

June 4, 2019

Contents

1 Weighted K means algorithm

According to the slides and paper (Chun and Yuen, 2013), create R code chunks and complete the codes step by step as follows. At the same time, pratice your skills to use R Markdown.

1.1 Step 1: Data matrix normalization

1.1.1 Data Source.

We create a data matrix for this case. hints: use *matrix()*.

1.2 Declare the related variables

We define the related variables as below.

```
#K: number of clusters
(K=2)
#m: number of rows of dataset X
(m=nrow(X))
#Number of features of dataset X
(n=ncol(X))
#P: cluster result pattern
(P=vector(mode = "numeric",length = nrow(X)));
#dcDist: distance from a data point to a closest clusters
(dcDist=vector(mode = "numeric",length = nrow(X)));
```

1.3 Normalization function

There are different ways to normalize the data. one of the methods is shown as below.

$$\hat{x}_{ij} = \frac{x_{ij} - \min x_{kj}}{\max x_{kj} - \min x_{kj}}, k \in \{1, 2, ..., m\}$$

Create a normalization function and test it.

Next,we can use *apply* function for a matrix where MARGIN = 1 indicates rows, MARGIN = 2 indicates columns. This will create a normalized data.

2 Step 2: Attributes' weights determination

Classical K-means method assumes each criterion has the equal weight, but this may not reflect the decision maker's preference. Cognitive pairwise comparison can adjust the weight of the criteria /variables, which will be introduced in the future. We simply define the weights as below.

```
W= c(0.32,0.28,0.40)
```

3 Step 3: data partition by k-means approach

3.1 Choose centroids

in this case, we need to classify the above members into two clusters, A and B.

Supposed ID2 is chosen for Cluster A, and ID6 is chosen for Cluster B.

3.2 Compute the Euclidean distances

3.2.1 Process to develop the function

Compute the weighted Euclidean distances from objects to all the centers by

$$d_{ij} = \sqrt{w_1|x_{i1} - x_{j1}|^2 + w_2|x_{i2} - x_{j2}|^2 + \cdots + w_n|x_{in} - x_{jn}|^2}$$

We take ID2, i.e. the 1st row, i=1, as example. You can change the i value to check the other data object.

3.2.2 Use *for* loop

We pack the calculation steps above into a function. Assign each object to the cluster of minimum distance from this object to the cluster center

for the initial round, {2,3} is one cluster, and {1,4,5,6}.

3.2.3 Update mean values

Calculate the mean of each cluster, then update the centers of clusters by the mean values.

3.2.4 Use *for* loop for the step above

We pack the steps above as the general function by looping.

3.3 Re-do again

We repeat the early steps again, i.e. copy the code in the early step.

We found the {2,3} is one group, whilst {1,4,5,6} is one group. The result is the same as the previous step. if we recalculate the mean, would be the same as the previous one? If yes, we can stop the process.

3.3.1 Question

Arrange the function, pack the functions above as a general k-means function.

4 The Limitation of k-means

In this lab, we create a toy example to check the limitation of k-means.

4.1 Data Matrix

Create the toy dataset.

```
C1=c(0.1,0.5,0.2,1,0.8,0.3)
C2=c(0.4,0.3,0,0.8,0.3,0.1)
X=matrix(data=c(C1,C2),6,2)
X
```

Create a plot for the data matrix above.

4.2 Perform R build-in Kmeans function

Remember set `algorithm ="MacQueen"` for kmeans.

```
cluster=kmeans(X,2, algorithm ="MacQueen" )
cluster
```

4.3 Simulation

We run kmean for the same dataset for 10 times. We print the label results and plot the graphs.

```
for(i in 1:10)
{
  cat("Iteration ", i, ": \n")
  cluster=kmeans(X,2, algorithm ="MacQueen")$cluster
  names(cluster) =1:6
  print(cluster)
 # plot(C1,C2,col=cluster,main=i)
  plot(X, col="white")
  text(C1,C2, col=cluster, labels = 1:6)
}
```

4.4 Question

According to the clustering results, what will you conclude?

5 Hierarchical Clustering methods

We skip the concepts of Hierarchical Clustering. We simply apply the functions to handle the same dataset.

```
###HC####
dist.X=dist(X)
k=2

#Complete Linkage
hc.complete=hclust(dist(X), method="complete")
plot(hc.complete,main="Complete Linkage", xlab="", sub="", cex=.9)
hc.complete
cutree(hc.complete, k = k)

#Single Linkage
hc.single=hclust(dist(X), method="single")
plot(hc.single,main="Single Linkage", xlab="", sub="", cex=.9)
hc.single
cutree(hc.single, k = k)

#Average Linkage
hc.average=hclust(dist(X), method="average")
plot(hc.average,main="Average Linkage", xlab="", sub="", cex=.9)
hc.average
cutree(hc.average, k = k)
```

6 Spectral Clustering

```
library("kernlab")
data(spirals)
X=spirals
rownames(X)=1:nrow(X)
sc <- specc(X, centers=2)
sc
centers(sc)
size(sc)
withinss(sc)
plot(X, col=sc)
```

7 Comparisons

We explore the package, `clusterCrit`,

```
#bestCriterion {clusterCrit}
library(clusterCrit)
example(extCriteria)
example(intCriteria)
```

We consider the methods as below

- **intCriteria**: "Dunn","Silhouette";
- **extCriteria**: "Jaccard", "Precision","Rand".

Compare k-means, 3 types of Hierarchical Clustering methods, and Spectral Clustering for *iris* dataset, a build-in R dataset, and *spirals* dataset in *kernlab* package.

```
head(iris)
```

8 References

Guan, C., and Yuen K.K.F., (2013) "Toward A Hybrid Approach of Primitive Cognitive Network Process and K-Means Clustering for Social Network Analysis", Proceedings of 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp.1267- 1271.