­

Ch10. Unsupervised Learning

-mclust pckg & Lab & Exercises #7,10

­­

**학번 : 182STG12**

**이름 : 오혜윤**

c

Problem Description

In mclust tutorial, we learn about the details of mclust functions. There are two packages which provide mclust modeling functions. First, we can use the Mclust() function and there are several useful parameters we can choose in Mclust() function. The parameter we mostly use is BIC. In this tutorial, we learn about modifying the options of mclust functions and choosing the best classification model for the dataset.

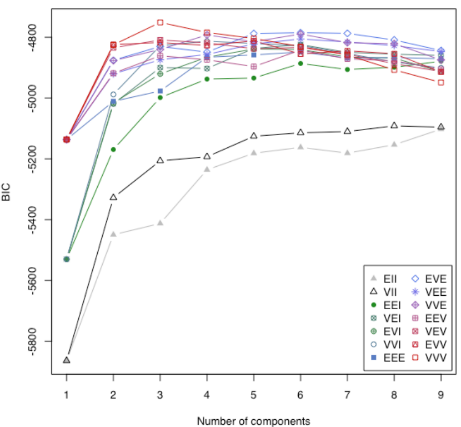
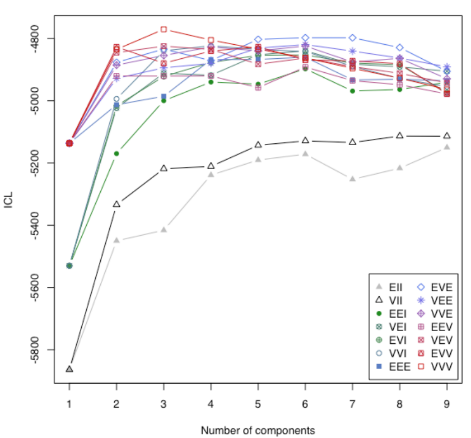
In Chapter 10 lab, we learn about PCA and clustering. The function “kmeans” and “hclust” are commonly used for clustering. We can modify the number of clusterings and choose the best model.

In exercises 7 and 10, using random dataset, we apply various clustering methods. We do PCA or scaling before we do Kmeans clustering. Also, we can modify the options of kmeans function. As a result, we choose the best K for the besting clustering model.

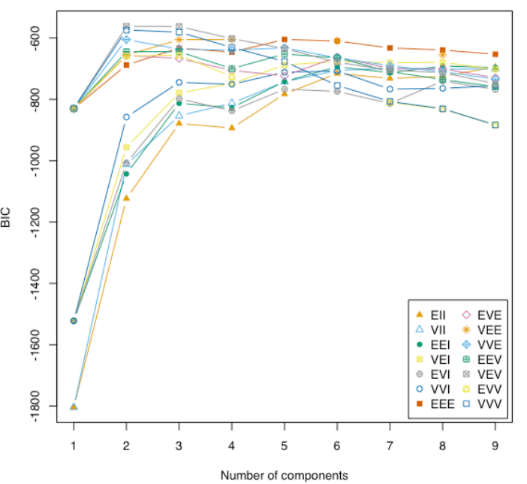
Results

**mclust Tutorial Review**

In mclust tutorial, we learn about the Mclust() function. “mclust” is a contributed R package for model-based clustering, classification, and density estimation based on finite normal mixture modeling. In this package, we can use the Mclust() function and there are several useful parameters we can choose in this function. The parameters we mostly use are BIC or ICL. We compare the BICs and choose the model with less BIC. Also, we can update BIC by merging the best results.



There are other functions including MclustBootstrap() and densityMclust(). We use these functions and choose the number of components of the best model. The plot below represents the iris dataset classification.



**CH10. Lab Review**

In this lab, it was mostly about PCA and clustering. The function “kmeans” and “hclust” are commonly used for clustering. We can modify the number of clusterings and choose the best model.

Using NCI60 dataset which consists of cancer cell line microarray data, we do PCA and clustering. As a result of PCA, we can conclude that cell lines from the same cancer type tend to have pretty similar gene expression levels. Also, we do hclustering and we can conclude that Complete and average linkage tend to yield evenly sized clusters whereas single linkage tends to yield extended clusters to which single leaves are fused one by one.

**CH10 Exercises**

**7. In the chapter, we mentioned the use of correlation-based distance and Euclidean distance as dissimilarity measures for hierarchical clustering. It turns out that these two measures are almost equivalent: if each observation has been centered to have mean zero and standard**

**deviation one, and if we let rij denote the correlation between the ith and jth observations, then the quantity 1−rij is proportional to the squared Euclidean distance between the ith and jth observations. On the USArrests data, show that this proportionality holds.**

|  |
| --- |
| R codes: |
| library(ISLR)  set.seed(1)  dsc <- scale(USArrests)  d1 <- dist(dsc)^2 #euclidean  d2 <- as.dist(1 - cor(t(dsc))) #correlation based  summary(d2 / d1) |

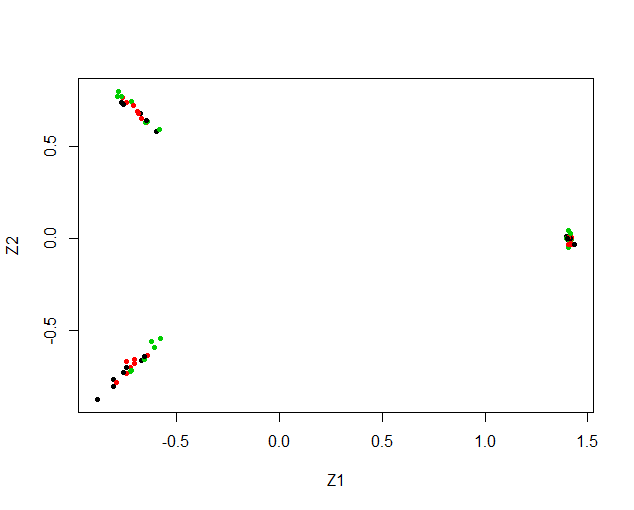
You can see the result below. We can conclude that this proportionality is true.

|  |
| --- |
| Results: |
| Min. 1st Qu. Median Mean 3rd Qu. Max.  0.000086 0.069135 0.133943 0.234193 0.262589 4.887686 |

**10. In this problem, you will generate simulated data, and then perform PCA and K-means clustering on the data.**

**(a) Generate a simulated data set with 20 observations in each of three classes (i.e. 60 observations total), and 50 variables.**

|  |
| --- |
| R codes: |
| set.seed(2)  x<-matrix(rnorm(20\*3\*50,mean=0,sd=0.1),ncol=50)  x[1:20,2]<-1  x[21:40,1]<-2  x[21:40,2]<-2  x[41:60,1]<-1  true.labels<-c(rep(1,20),rep(2,20),rep(3,20)) |

 **(b) Perform PCA on the 60 observations and plot the first two principal component score vectors. Use a different color to indicate the observations in each of the three classes. If the three classes appear separated in this plot, then continue on to part (c). If not, then return to part (a) and modify the simulation so that there is greater separation between the three classes. Do not continue to part (c) until the three classes show at least some separation in the first two principal component score vectors.**

**(c) Perform K-means clustering of the observations with K = 3. How well do the clusters that you obtained in K-means clustering compare to the true class labels?**

|  |  |  |  |
| --- | --- | --- | --- |
| **true.labels** | **1** | **2** | **3** |
| **1** | 20 | 0 | 0 |
| **2** | 0 | 20 | 0 |
| **3** | 0 | 0 | 20 |

There were 3 classes in original data set with 20 observations in each. We can conclude that the dataset was perfectly clustered in 3 classes obtained from k-means clustering with K=3.

**(d) Perform K-means clustering with K = 2. Describe your results.**

|  |  |  |
| --- | --- | --- |
| **true.labels** | **1** | **2** |
| **1** | 20 | 0 |
| **2** | 0 | 20 |
| **3** | 20 | 0 |

The original 3 classes are now clustered into 2 only. 3 cluster is clubbed with another completely.

**(e) Now perform K-means clustering with K = 4, and describe your results.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **true.labels** | **1** | **2** | **3** | **4** |
| **1** | 0 | 11 | 0 | 9 |
| **2** | 20 | 0 | 0 | 0 |
| **3** | 0 | 0 | 20 | 0 |

The original 3 classes are now classified into 4 clusters. With 2 classes perfectly classified into 2 separate clusters but observations of one class is distributed over 2 clusters.

**(f) Now perform K-means clustering with K = 3 on the first two principal component score vectors, rather than on the raw data. That is, perform K-means clustering on the 60 × 2 matrix of which the first column is the first principal component score vector, and the second column is the second principal component score vector. Comment on the results.**

|  |  |  |  |
| --- | --- | --- | --- |
| **true.labels** | **1** | **2** | **3** |
| **1** | 0 | 0 | 20 |
| **2** | 0 | 20 | 0 |
| **3** | 20 | 0 | 0 |

All observations are perfectly clustered.

**(g) Using the scale() function, perform K-means clustering with K = 3 on the data after scaling each variable to have standard deviation one. How do these results compare to those obtained in (b)? Explain.**

|  |  |  |  |
| --- | --- | --- | --- |
| **true.labels** | **1** | **2** | **3** |
| **1** | 3 | 15 | 2 |
| **2** | 0 | 0 | 20 |
| **3** | 12 | 6 | 2 |

Here we see the observations are not perfectly clustered and the results are worse than unscaled clustering. Scaling affects the distance between the observations and hence should be avoided till mandatory

river?

Discussion

We applied various PCA and clustering approaches to many models. In the datasets we used, the tuned Kmeans model was not the best in all cases. So, we can conclude that the best models fitting to the dataset can be different in many. Also, we can use scaling or PCA for clustering. We should try several models with different nstart and k. Through this process, we can choose the best classification model for the dataset we used.

Appendix (R

**R codes:**

###ex7---------------------------------------------------------------------------

library(ISLR)

set.seed(1)

dsc <- scale(USArrests)

d1 <- dist(dsc)^2#euclidean

d2 <- as.dist(1 - cor(t(dsc)))#correlation based

summary(d2 / d1)

###ex10---------------------------------------------------------------------------

##a)---------------------------------------------------------------------

set.seed(2)

x<-matrix(rnorm(20\*3\*50,mean=0,sd=0.1),ncol=50)

x[1:20,2]<-1

x[21:40,1]<-2

x[21:40,2]<-2

x[41:60,1]<-1

true.labels<-c(rep(1,20),rep(2,20),rep(3,20))

##b)---------------------------------------------------------------------

pca<-prcomp(x)

plot(pca$x[,1:2],col=1:3,xlab="Z1",ylab="Z2",pch=20)

##c)---------------------------------------------------------------------

km.clustering <- kmeans(x, 3, nstart = 20)

table(true.labels, km.clustering$cluster)

##d)---------------------------------------------------------------------

km.clustering.2 <- kmeans(x, 2, nstart = 20)

table(true.labels, km.clustering.2$cluster)

##e)---------------------------------------------------------------------

km.clustering.4<- kmeans(x, 4, nstart = 20)

table(true.labels, km.clustering.4$cluster)

##f)---------------------------------------------------------------------

km.pca<-kmeans(pca$x[,1:2],3,nstart=20)

table(true.labels,km.pca$cluster)

##g)---------------------------------------------------------------------

kmscale<-kmeans(scale(x),3,nstart=20)

table(true.labels,kmscale$cluster)