Data Mining

HW#9: Unsupervised Learnings

학번	182STG18
이름	이하경
제출일	2019.05.22

Description

Unsupervised Learnings

Regression 이나 Classification 에 사용하는 많은 방법론들은 모두 주어진 관측치의 p 개의 특징변수 $X_1, \ldots, X_p$의 값과 쌍으로 주어지는 반응변수 Y 의 값을 예측하는 데 사용된다. 이렇게 반응변수를 예측하는 알고리즘을 지도학습(Supervised Learning)이라고 한다. 이와 다르게 본 과제에서 다룰 비지도학습(Unsupervised Learning) 방법론들은 반응변수 Y 없이 특징변수 $X_1, \ldots, X_p$ 만을 이용해 특정 변수나 관측치들 사이의 관계 등 정보를 얻는 것이다. 비지도 학습은 예측과 같은 명확한 목표가 없기 때문에 비교적 주관적이지만 Y 가 존재하지 않는 실생활의 많은 데이터에 대해 적용 가능하다. 본 과제에서 다루는 비지도학습 방법론은 크게 다음의 두 가지이다.

주성분 분석 (Principal Component Analysis, PCA)

PCA 는 데이터의 차원을 줄이는 방법으로 p 개의 설명변수에서 가능한 p 개의 선형결합을 찾는다. 새롭게 만들어진 p 개의 선형결합, 즉 주성분은 주성분 1 부터 차례대로 설명변수 전체의 분산 중 가장 큰 분산 비율을 차지하면서 서로 상관관계가 없도록 만들어진다. PCA 는 지도학습에서 p 개의 설명변수에서 차원의 축소 효과를 얻을 뿐만 아니라 설명변수가 많은 경우의 시각화에도 용이하게 사용된다.

클러스터링 (Clustering)

클러스터링은 주어진 관측치들을 몇 개의 군집(subgroup)으로 구분하는 방법이다. 전체 관측치들의 공간을 설명변수들의 특성이 비슷한 k 개의 공간으로 분할함으로써 같은 군집에 속한 관측치들끼리는 가능한 한 서로 동질적이며 서로 다른 군집의 관측치들은 서로 동질적이지 않기를 기대한다. 관측치 간 '유사도'는 주로 거리로 나타낼 수 있지만 이에 대한 명확한 정의는 없으므로, 이것을 어떻게 정의하고 해석하는 지에 따라 같은 표본에서도 클러스터링의 결과가 다를 수 있으며 하나의 뚜렷한 정답은 없다. 클러스터링은 크게 모델 기반 클러스터링(Model-based Clustering), 즉 관측치들의 분포를 가정한 클러스터링과 분포 가정 없이 관측치 간 유사도 및 거리를 판단하는 클러스터링으로 구분할 수 있다.

본 과제에서는 대표적인 모델 기반 클러스터링인 GMM(Gaussian Mixture Model)을 R 에서 실행해보고, ISLR 의 연습문제에서는 K-means 클러스터링을 적용해보았다.

Results

A quick tour of 'mclust'

mclust 는 R 에서 Gaussian Mixture 모델에 기반한 클러스터링 및 분류, 밀도함수 추정을 수행할 수 있는 R 패키지이다. 변수들의 공분산 구조 가정에 따른 다양한 방법으로 Gaussian Mixture 모형에 EM 알고리즘을 적용하여 모수를 추정할 수 있으며, 모형으로부터 시뮬레이션(Bootstrap)을 하는 것도 가능하다. 또한 여러 모형 중 최적의 모형을 선택하기 위한 BIC 를 비교할 수 있다. 적합한 모형의 summary 와 plot 을 통해 결과를 수치적, 시각적으로 요약할 수 있다. 튜토리얼을 통해 모델 기반 클러스터링(GMM)을 R 에서 diabetes, galaxies, iris 등의 데이터에 직접 적용하고 다양한 함수를 사용해 결과를 출력해볼 수 있었다.

Chapter 10 Lab

연습문제를 풀기 전 PCA, K-means 클러스터링과 계층적 클러스터링을 R 에서 수행할 수 있는 함수들을 먼저 Lab 을 통해 학습하였다. USArrests 는 미국 50 개의 주에 대해 3 개의 범죄 체포율 변수와 도심 인구의 비율 변수가 존재하는

데이터 셋으로 이 데이터에 PCA 를 적용한 결과 첫번째 주성분이 전체 분산의 반 이상인 62%를 차지하는 것을 확인할 수 있었다.

클러스터링을 위해서는 정규분포로부터 시뮬레이션 샘플을 생성하여 kmeans 함수와 hclust 함수를 이용해 각각 군집화하였다. 또한 NCI60 데이터에 대해서도 PCA 와 두 가지 클러스터링을 직접 적용해보았다. 비지도학습의 모형들은 Y 를 사용하지 않고 주어진 설명변수만을 사용하므로, 설정한 옵션에 따라 결과가 달라질 수 있음을 확인하였다.

Exercise 10.7 USArrests Data

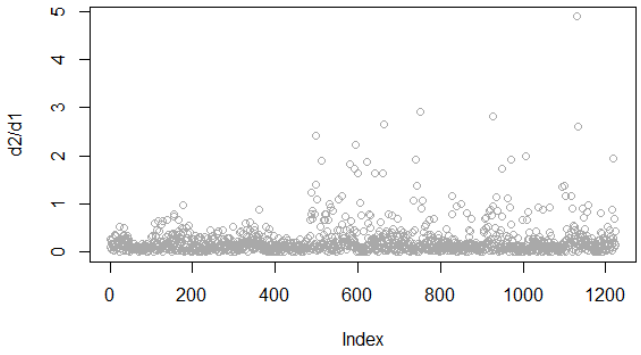
Correlation-based distance and Euclidean distance are used as dissimilarity measures for hierarchical clustering. These two measures are almost equivalent: if each observation has centered to have mean 0 and standard deviation 1, and if we let r_{ij} denote the correlation between the ith and jth observations, then the quantity $1 - r_{ij}$ is proportional to the squared Euclidean distance between the ith and jth obserbations.

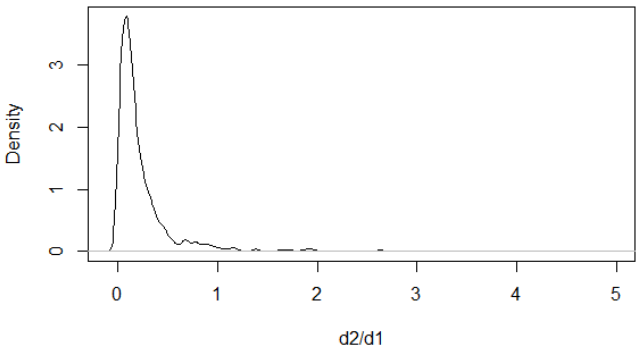
Show that tis proportionality holds on the USArrests data.

$$\text{Squared Euclidean distance } (d_1): \sum_{k=1}^{p} (x_{ik} - x_{jk})^p$$

$$\text{Correlation based distance } (d_2): \quad 1 - r_{ij}$$

USArrest 데이터에 있는 50 개의 관측치들 간의 거리를 두가지 방법으로 구하고 서로 비례하는지 확인하기 위해 d_2/d_1의 분포를 그래프 및 summary 통계량으로 확인하였다.





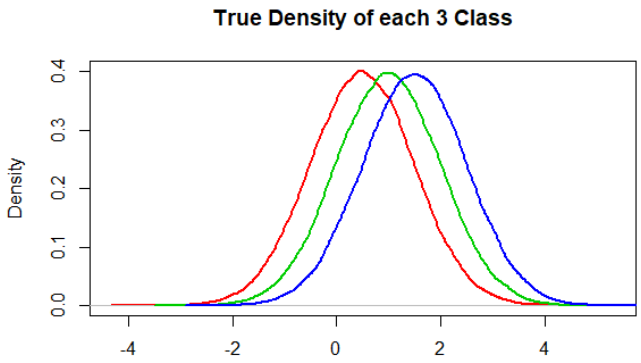
Min	Q1	Median	Mean	Q3	Max
0.0001	0.0697	0.1339	0.2342	0.2626	4.8877

일부 큰 값을 제외하고 두 값의 비례상수가 대부분 중앙값인 0.1339 주변에 분포한다. 따라서 두 가지 거리 척도에 비례관계가 성립하고, 서로 동등한 의미를 가진다고 할 수 있다.

Exercise 10.10 PCA and K-means clustering on a simulated data

(a) Generate a simulated data set with 20 observations in each of 3 classes (i.e. 60 observations total), and 50 variables.

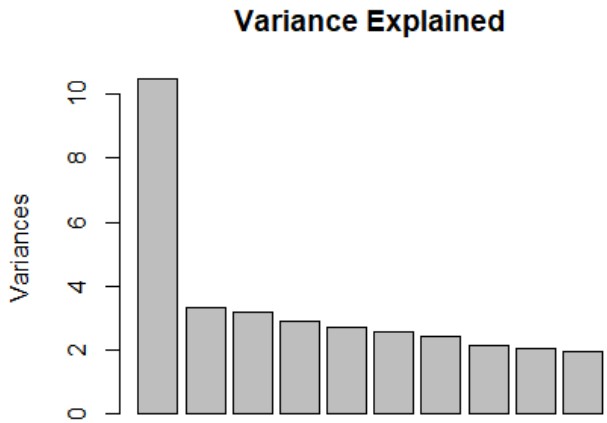
각 클래스에는 20 개의 관측치가 있고 관측치 하나당 가지는 특징의 개수(p)는 50 개이므로, 세 개의 클래스를 구성하기 위해 평균이 서로 다른 세 가지 정규분포로부터 각각 20*50 개씩의 랜덤한 값을 추출하였다.

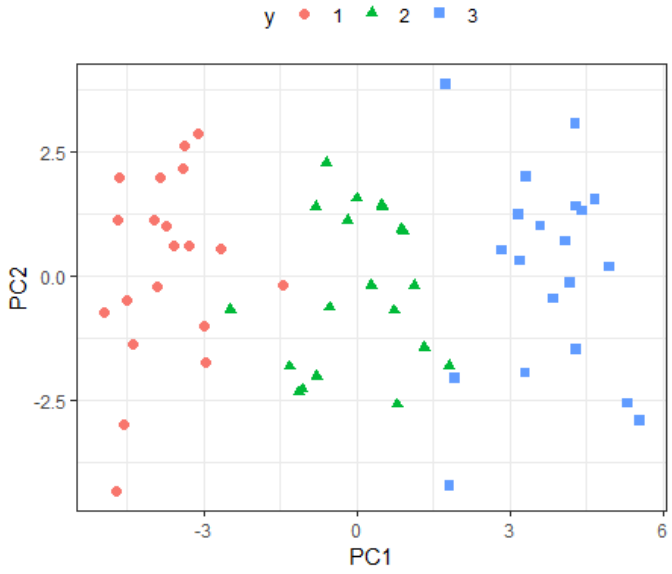


class 1	class 2	class 3
Normal(0.5, 1)	Normal(1, 1)	Normal(1.5, 1)

(b) Perform PDA on the 60 observations and plot the first two principal component score vectors.

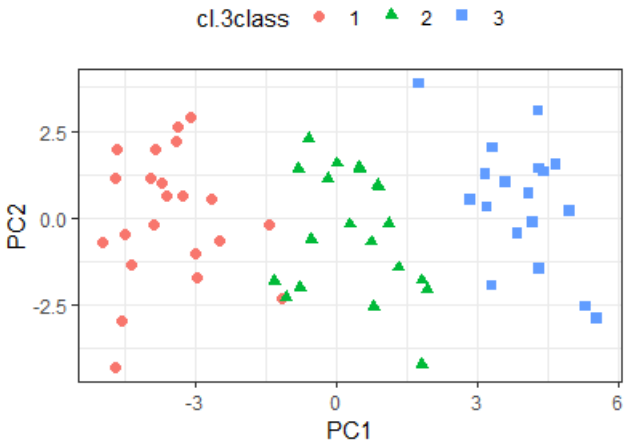
▷ **PCA Result**





처음 10 개 주성분의 분산을 나타내는 그래프를 볼 때 첫번째 주성분이 전체 분산의 대부분을 차지하고 있음을 알 수 있다. 주성분 1 과 주성분 2 의 score 를 각각 x, y 축으로 하여 각 그룹에 따라 두 개의 주성분 값의 차이를 확인하였을 때 주성분 2 는 그룹에 따라 차이가 없으나 주성분 1 은 경계의 한 두개의 관측치를 제외하고는 그룹 간 차이가 뚜렷하였다.

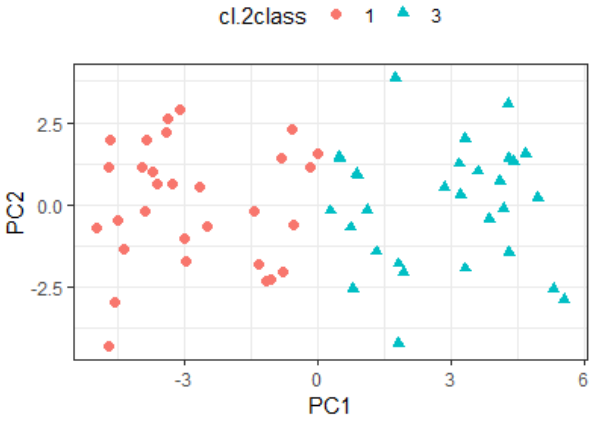
(c) Perform K-means clustering of the observations with K=3. How well do the clusters that you obtained in K-means clustering compare to the true class labels?



	CLUSTER		
TRUE	**1**	**2**	**3**
1	20	0	0
2	2	18	0
3	0	2	18

관측치 하나당 p=50 개의 모든 값을 사용해 K=3 인 k-means 클러스터링을 하였을 때 경계에서 각각 두 개씩의 관측치를 제외하고 나머지 관측치들은 모두 원래의 그룹과 동일하게 그룹화되었다. 즉 총 60 개 중 4 개의 관측치만이 잘못된 그룹에 속한다.

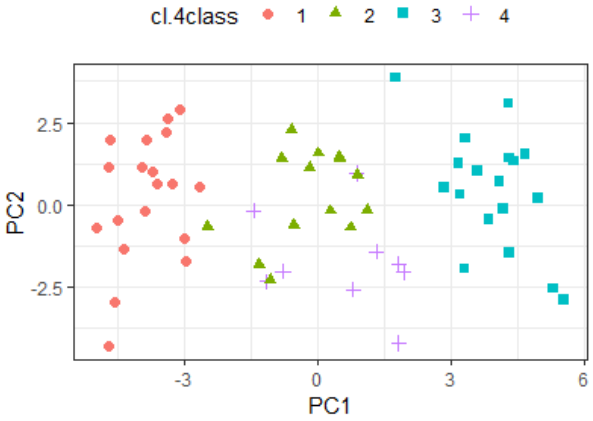
(d) Perform K-means clustering with K=2.



	CLUSTER		
TRUE	**1**	**2**	**3**
1	20	0	0
2	10	0	10
3	0	0	20

관측치들을 k-means 클러스터링으로 2 그룹으로 나누었을 때 주성분을 이용해 산점도를 그려보면 그룹 2 의 관측치들이 모두 그룹 1 과 3 에 10 개씩 나누어 그룹화된 것을 확인할 있다. nstart=100 으로 설정한 결과 여러 번 k-means 를 수행해도 결과가 동일하였다. 60 개의 관측치를 두 그룹으로 분리하기 위해서는 알고리즘이 적절하게 분리하였다고 할 수 있다.

(e) Perform K-means clustering with K=4.

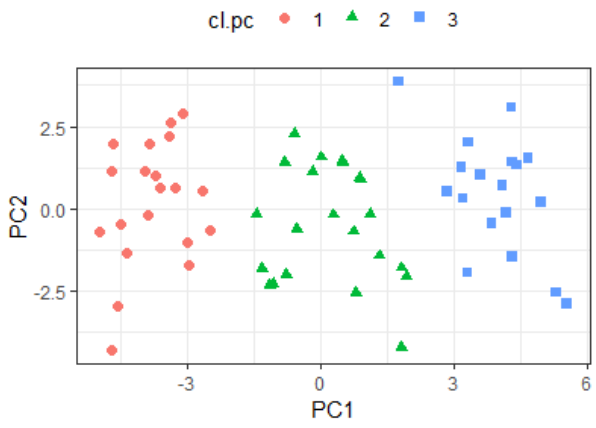


	CLUSTER			
TRUE	**1**	**2**	**3**	**4**
1	19	0	0	1
2	0	14	0	6
3	0	0	18	2

그룹의 수를 실제 그룹보다 많게 해 K=4 인 k-means 클러스터링을 하였을 때 실제 그룹 1 과 3 의 관측치들은 서로 거의 같은 그룹으로 묶였으나 그룹 2 의 관측치 중 일부가 다른 그룹으로 분류되었다. nstart=100 으로 한 결과 클러스터링 결과가 동일하였다.

(f) Perform K-means clustering with K=3 on the first 2 Principal Component Score vectors, rather than on the raw data.

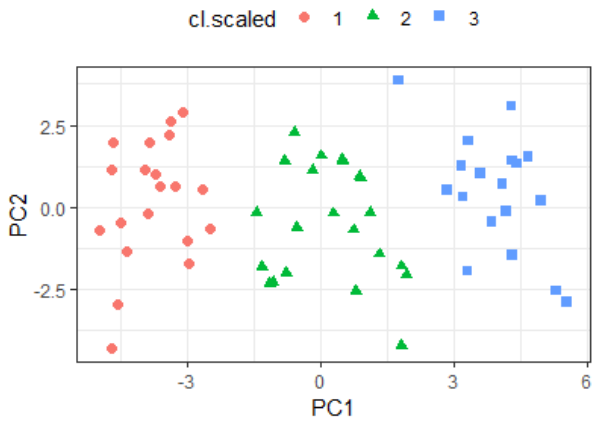
관측치별 주성분 1 과 주성분 2 의 Score 만을 이용해 K=3 인 k-means 클러스터링을 다시 진행하였다.



	CLUSTER		
TRUE	**1**	**2**	**3**
1	19	1	0
2	1	19	0
3	0	2	18

실제 그룹 간에도 값의 차이가 확연한 주성분 1 의 경우 3 가지 그룹으로 뚜렷하게 구분되었다. 3 가지 관측치를 제외하고 실제 그룹과 거의 동일하게 분류하는 것을 확인할 수 있다.

(g) Perform k-means clustering with K=3 on the data after scaling each variable to mean 0 and sd 1.



	CLUSTER		
TRUE	**1**	**2**	**3**
1	19	1	0
2	1	19	0
3	0	2	18

관측치별 p 개의 변수 값을 평균 0, 분산 1 으로 정규화하고 50 개의 변수를 모두 사용해 세 그룹으로 클러스터링하였다. 정규화를 한 결과 하기 전의 결과인 (b)와 일부 관측치를 분류한 그룹이 다르지만 전체적으로 실제 클래스를 잘못 분류한 관측치의 개수는 4 개로 동일하다.

Discussion

클러스터링에서 관측치 간 거리를 결정하는 방법은 설명변수가 모두 수치형일 때 주로 유클리디안 거리를 사용한다. 거리를 척도로 이용하는 방법론에서 주의할 점은 설명변수들 사이의 분포의 차이가 클 경우 거리를 측정하는 데에 변수별로 미치는 영향의 크기가 다를 수 있다는 것이다. 따라서 거리를 측정할 때는 각각의 설명변수들의 평균과 분산이 동일하도록 데이터를 정규화하여 계산하는 것이 좋다. 연습문제 10.7 에서는 유클리디안 거리와 상관계수 기반의 거리의 관계를 USArrest 데이터를 통해 확인하였다.

연습문제 10.7 에서는 p=50 으로 특징변수가 상당히 많은 경우를 가정하여 시뮬레이션된 데이터를 이용하였다. 3 가지 서로 다른 평균을 가진 정규 분포에서 각각 20 개씩의 관측치를 생성하였으므로 실제 클래스 Y 의 값이 1, 2, 3 으로 구분될 수 있지만 K-means 클러스터링을 통해 Y 없이 60 개의 관측치들을 K=3, K=2, K=4 로 설정하였을 때 어떻게 군집화되는지 확인하였다. 설명변수가 매우 많아 관측치들의 분포를 시각적으로 확인하기 어려우나 PCA 를 활용해 분산이 가장 큰 첫번째와 두번째 주성분만으로 산점도를 그림으로써 세 그룹간 주성분의 차이를 확인할 수 있었다. 따라서 PCA 가 데이터의 차원 축소 뿐만 아니라 시각화에도 용이하게 사용할 수 있음을 깨달았다.

[Appendix] R code

Exercise 10.7

```
data(USArrests)
data.scaled <- scale(USArrests)
d1 <- dist(data.scaled)^2
d2 <- as.dist(1-cor(t(data.scaled)))
summary(d2/d1)

plot(d2/d1, col = 'darkgray')
plot(density(d2/d1), main = '', xlab = 'd2/d1')
```

Exercise 10.10

```
# (a)
n <- 20 ; p <- 50
set.seed(1010)
X <- rbind(
  matrix(rnorm(n*p, mean = 1*0.5), n, p),
  matrix(rnorm(n*p, mean = 2*0.5), n, p),
  matrix(rnorm(n*p, mean = 3*0.5), n, p)
)
y <- as.factor(rep(1:3, each = n))

plot(density(rnorm(100000,1*0.5)), col = 'red1', lwd = 2,
     main = 'True Density of each 3 Class', xlab = '')
lines(density(rnorm(100000,2*0.5)), col = 'green3', lwd = 2)
lines(density(rnorm(100000,3*0.5)), col = 'blue1', lwd = 2)

# (b)
summary(pc.out <- prcomp(X))
plot(pc.out, main = 'Variance Explained')

pc.two <- as.data.frame(pc.out$x[,1:2])
ggplot(pc.two) + theme_bw() + theme(legend.position = 'top') +
  geom_point(aes(PC1, PC2, shape = y, color = y), size = 2)

# (c)
set.seed(10)
(km.3class <- kmeans(X, 3, nstart = 100))
cl.3class <- factor(c(3:1)[km.3class$cluster], levels = c(1:3))

table(true = y, kmeans = cl.3class)
ggplot(pc.two) + theme_bw() + theme(legend.position = 'top') +
  geom_point(aes(PC1, PC2, shape = cl.3class, color = cl.3class), size = 2)

# (d)
set.seed(10)
(km.2class <- kmeans(X, 2, nstart = 100))
cl.2class <- factor(c(3,1)[km.2class$cluster], levels = c(1:3))

ggplot(pc.two) + theme_bw() + theme(legend.position = 'top') +
  geom_point(aes(PC1, PC2, shape = cl.2class, color = cl.2class), size = 2)
table(true = y, kmeans = cl.2class)

# (e)
set.seed(10)
(km.4class <- kmeans(X, 4, nstart = 100))
cl.4class <- factor(c(2,1,3,4)[km.4class$cluster], levels = c(1:4))

ggplot(pc.two) + theme_bw() + theme(legend.position = 'top') +
  geom_point(aes(PC1, PC2, shape = cl.4class, color = cl.4class), size = 2)
table(true = y, kmeans = cl.4class)

# (f)
set.seed(10)
(km.pc <- kmeans(pc.two, 3, nstart = 100))
cl.pc <- as.factor(c(3,1,2)[km.pc$cluster])

ggplot(pc.two) + theme_bw() + theme(legend.position = 'top') +
  geom_point(aes(PC1, PC2, shape = cl.pc, color = cl.pc), size = 2)
table(true = y, kmeans = cl.pc)

# (g) compare to (b)
Xsc <- scale(X)
set.seed(10)
(km.scaled <- kmeans(Xsc, 3, nstart = 100))
cl.scaled <- factor(c(1,3,2)[km.scaled$cluster], levels = c(1:3))
ggplot(pc.two) + theme_bw() + theme(legend.position = 'top') +
  geom_point(aes(PC1, PC2, shape = cl.scaled, color = cl.scaled), size = 2)
table(true = y, kmeans = cl.pc)
```