**###補充教材--- Unit 2: 分群與關聯 (非監督式)**

install.packages("MASS")

library(MASS)

install.packages("cluster")

library(cluster)

install.packages("clusterSim")

library(clusterSim)

data(iris)

iris.nm<-data.Normalization (iris[, -5], type="n4", normalization="column")

gender\_size<- read.csv("[gender\_size.csv](file:///\\gender_size.csv)")

gender.nm<-data.Normalization (gender\_size[, -4], type="n4", normalization="column")

min.nc=2

max.nc=8

K means

result0= kmeans(iris[,1:4], center=3)

print(result0)

table(iris$Species, result0$cluster)

plot(iris[,1:2], pch=(result0$cluster), col=result0$cluster)

points(result0$centers, col=1:3, pch=8)

KM=array(0, c(max.nc-min.nc+1,2))

for (nc in min.nc : max.nc)

{ fitkm=kmeans(iris[, -5], center=nc)

KM[nc-(min.nc-1), 1]= fitkm$betweenss/fitkm$tot.withinss

KM[nc-(min.nc-1), 2]= index.DB(iris[,-5], fitkm$cluster, centrotypes="centroids", p=2)$DB

}

which(KM[,1]==max(KM[,1]))

which(KM[,2]==min(KM[,2]))

install.packages("NbClust")

library(NbClust)

NbClust(iris.nm, distance="euclidean", min.nc=2, max.nc=8, method="kmeans", index="all")

K medoids

PM=array(0, c(max.nc-min.nc+1,2))

for (nc in min.nc : max.nc)

{ fitpm=pam(iris[,-5], nc)

PM[nc-(min.nc-1), 1]= index.DB(iris[,-5], fitpm$clustering, centrotypes="centroids", p=2)$DB

PM[nc-(min.nc-1), 2]= index.DB(iris[,-5], fitpm$clustering, dist(iris[,-5]), centrotypes="medoids", p=2)$DB

}

which(PM[,1]==min(PM[,1]))

which(PM[,2]==min(PM[,2]))

result1=pam(iris[,-5], 2)

print(result1)

table(iris$Species, result1$cluster)

plot(iris[,1:2], pch=(result1$cluster), col=result1$cluster)

points(result1$centers, col=1:3, pch=8)

C means

install.packages("e1071")

library(e1071)

CM= array(0, c(max.nc-min.nc+1, 2))

for (nc in min.nc : max.nc)

{ fitcm=cmeans(iris[,-5], centers=nc, m=2, verbose=TRUE, method="cmeans")

CM[nc-(min.nc-1), 1]= fclustIndex(fitcm, iris[,-5], index="xie.beni")

CM[nc-(min.nc-1), 2]= fclustIndex(fitcm, iris[,-5], index="fukuyama.sugeno")

}

which(CM[,1]==min(CM[,1]))

which(CM[,2]==min(CM[,2]))

result2<- cmeans(iris[,-5], center=2, m=2, iter.max=200, verbose=T, method="cmeans")

print(result2)

table(iris$Species, result2$cluster)

Hierarchical clustering

id=sample(1:nrow(iris), 0.2\*nrow(iris))

result3=hclust(dist(iris[id, -5]), method="average")

print(result3)

plot(result3, labels=iris$Species[id])

group3=cutree(result3, k=3)

print(group3)

table(group3)

rect.hclust(result3, k=3, border="red")

HC= array(0, c(max.nc-min.nc+1,2))

for (nc in min.nc : max.nc)

{

fithc1 <- hclust(dist(iris[, -5]), method="average")

ct1 <- cutree(fithc1, k=nc)

HC[nc-(min.nc-1), 1] <- index.DB(iris[, -5], ct1, centrotypes="centroids")$DB

fithc2 <- hclust(dist(iris[, -5]), method="ward.D")

ct2 <- cutree(fithc2, k=nc)

HC[nc-(min.nc-1), 2] <- index.DB(iris[, -5], ct2, centrotypes="centroids")$DB

}

which(HC[,1]==min(HC[,1]))

which(HC[,2]==min(HC[,2]))

EM clustering

install.packages("mclust")

library(mclust)

result4=Mclust(iris[,-5], G=3)

summary(result4, parameters=TRUE)

names(result4)

result4$classification

fitEM1=Mclust(iris[,-5])

summary(fitEM1, parameters=TRUE)

names(fitEM1)

fitEM1$BIC

plot(fitEM1) # (1:BIC, 2:classification, 3:uncertainty, 4:density, 0:exit)

result5=Mclust(data\_gender[,-4], G=2)

summary(result5, parameters=TRUE)

fitEM2=Mclust(data\_gender[,-4])

summary(fitEM2, parameters=TRUE)

iris.BIC=mclustBIC(iris[,-5], G=seq(from=1, to=9, by=1))

iris.BIC

plot(iris.BIC)

iris.BICsum=summary(iris.BIC, data=iris[,-5])

iris.BICsum

names(iris.BICsum)

mclust2Dplot(iris[, 3:4], classification=iris.BICsum$classification)

iris.dens1=densityMclust(iris[, 3:4])

plot(iris.dens1, iris[, 3:4], col="blue", nlevels=15)

iris.dens2=densityMclust(iris[,1:2])

plot(iris.dens2, iris[, 1:2], type="persp", col="red")

\*\*\*再論clustering (NBA資料集)

nba<-read.csv("2011NBA.csv", header=T, sep=",")  
nba[1:5,]

nba.clus<-nba[,-c(1,2)] #排除資料集中的player與division文字欄位means<-apply(nba.clus, 2, mean) #計算欄變數的平均數，2表column，1表row  
sds<-apply(nba.clus, 2, sd) #計算每欄變數的標準差，2表column，1表row  
nba.clus<-scale(nba.clus, center=means, scale=sds) #標準化欄變數，計算Z分數  
nba.dist<-dist(nba.clus, method="euclidean") #計算歐幾里得距離  
nba.fit<-hclust(nba.dist, method="ward.D2") #以Ward法進行集群分析

names(nba.fit) #平均得分、平均籃板、平均助攻、單場火鍋、單場抄截 (原始)

plot(nba.fit, labels=nba$player, main="NBA TOP25") #繪製集群分析樹狀圖  
rect.hclust(nba.fit, k=5, border="red") #以紅線界定5個集群

cluster5<-cutree(nba.fit, k=5) #指定集群分析為5群  
nba$player[cluster5==1] #呼叫屬於第一集群的分析結果

sapply(unique(cluster5), function(cluster5)nba$player) #呼叫全部集群分析結果

nba.new<-cbind(nba, cluster5) #將集群分析產生的label與原資料合併

nba.new[1:5,]

nba.new$cluster5<-factor(nba.new$cluster5, levels=c(1:5), labels=c("scorer", "defender", "point-guard", "combo-guard", "power-forward")) #得分、防守、控衛、全能衛、大前鋒 (五個集群依序命名)  
table(nba.new$division, nba.new$cluster5) #比較NBA東西區的集群分布

**\*\*\*關聯規則Association rules/Titanic + Groceries資料集**

install.packages("arules")

library(arules)

data("Titanic")

titanic0<-as.data.frame(Titanic)

str(titanic0)

titanic<-NULL

for (i in 1:4)

{

titanic<-cbind(titanic, rep(as.character(titanic0[, i]), titanic0$Freq))

}

titanic<-as.data.frame(titanic)

names(titanic)<-names(titanic0) [1:4]

rule0<- apriori(titanic, parameter=list(minlen=2, supp=.005, conf=0.8), appearance=list(rhs=c("Survived=No", "Survived=Yes"), default="lhs"))

sorted0<- sort(rule0, by="lift")

inspect(sorted0)

rule1<- apriori(titanic, parameter=list(minlen=3, supp=.005, conf=0.8), appearance=list(rhs=c("Survived=Yes"), default="lhs"))

sorted1<- sort(rule1, by="confidence")

inspect(sorted1)

rule2<- apriori(titanic, parameter=list(minlen=3, supp=.005, conf=0.8), appearance=list(rhs=c("Survived=No"), default="lhs"))

sorted2<- sort(rule2, by="confidence")

inspect(sorted2)

**###零售資料集**

data("Groceries")

summary(Groceries)

inspect(Groceries[1:8])

rules0=apriori(Groceries, parameter=list(support=.005, confidence=0.6))

rules0

inspect(rules0)

rule.sup=sort(rules0, by="support")

inspect(rule.sup)

rule.con=sort(rules0, by="confidence")

inspect(rule.con)

rule.lif=sort(rules0, by="lift")

inspect(rule.lif)

rules1=apriori(Groceries, parameter=list (maxlen=2, supp=.0001, conf=.1), appearance=list (rhs="mustard", default="lhs"))

inspect(rules1)

**#output frequent iemsets & visualization**

items1=apriori(Groceries, parameter=list (minlen=2, maxlen=4, supp=0.01, target="frequent itemsets"), control=list(sort=-1)) #廣度優先搜尋 (交易代碼為key)

summary(items1)

inspect(items1[1:10])

items2=eclat(Groceries, parameter=list (minlen=2, maxlen=4, supp=0.01, target="frequent itemsets"), control=list(sort=-1)) #深度優先搜尋 (商品代碼為key)

summary(items2)

inspect(items2[1:10])

install.packages("arulesViz")

library(arulesViz)

rules2=apriori(Groceries, parameter=list(support=0.004, confidence=0.6))

plot(rules2)

plot(rules2, measure=c("support", "lift"), shading="confidence")

plot(rules2, method="grouped")

plot(rules2, interactive=TRUE)

**\*\*\*\*\*\*Sequential Rule Mining**

install.packages("Matrix")

install.packages("arules")

install.packages("arulesSequences")

library(Matrix)

library(arules)

library(arulesSequences)

zakidata<-read.csv("zaki.csv", header=T, sep=",")

zaki <- read\_baskets(con = "zaki.txt", info = c("sequenceID","eventID","SIZE"))

s1 <- cspade(zaki, parameter = list(support = 0.5), control = list(verbose = TRUE))

summary(s1)

as(s1, "data.frame")

s2 <- cspade(zaki, parameter = list(support = 0.5 , maxlen=2), control = list(verbose = TRUE)) #maxlen:最大交易次數限制

summary(s2)

as(s2, "data.frame")

s3 <- cspade(zaki, parameter = list(support = 0.5 , maxsize=2), control = list(verbose = TRUE)) #maxsize:單次交易最大品項數量限制

summary(s3)

as(s3, "data.frame")

s4 <- cspade(zaki, parameter = list(support = 0.5 , mingap=10), control = list(verbose = TRUE)) #mingap:每次交易事件最短發生間隔時間限制

summary(s4)

as(s4, "data.frame")

s5 <- cspade(zaki, parameter = list(support = 0.5 , maxgap=10), control = list(verbose = TRUE)) #maxgap:每次交易事件最大發生間隔時間限制

summary(s5)

as(s5, "data.frame")

s6 <- cspade(zaki, parameter = list(support = 0.5 , maxwin=10), control = list(verbose = TRUE)) #maxgap:最近兩個交易事件最大發生間隔時間限制

summary(s6)

as(s6, "data.frame")

s7 <- cspade(zaki, parameter = list(support = 0.5 , maxgap=20 , mingap=10), control = list(verbose = TRUE)) #各種參數可搭配使用

summary(s7)

as(s7, "data.frame")

as(subset(s1, size(x) < 3), "data.frame") #最大交易次數限制

as(subset(s1, size(x) > 1), "data.frame") #最小交易次數限制

as(subset(s2, x %ain% c("B","F")), "data.frame")

#要求s2規則只列出同時出現B、F項目的規則

as(subset(s2, x %ain% c("B", "F") & support > 0.5), "data.frame")

#要求s2規則只列出同時出現B、F項目的規則，並限制support值大小

############# #建立confidence規則前，需先須建立support規則

r1 <- ruleInduction(s1, confidence = 0.5, control = list(verbose = TRUE))

summary(r1) #使用上述的s1規則

as(r1, "data.frame")

#要求單次交易項目同時有B、F的規則顯示在左手邊(前交易)

as(subset(r1, lhs(x) %ain% c("B", "F")), "data.frame")

r2 <- ruleInduction(s2, confidence = 0.7, control = list(verbose = TRUE))

summary(r2) #使用上述的s2規則

as(r2, "data.frame")

#要求單次交易項目同時有A的規則顯示在右手邊(後交易)

as(subset(r2, rhs(x) %ain% c("A")), "data.frame")