INFS7203 REPORT

HEXIN ZHENG

4501283

Task1

1.1

#Extract data and assign a name to data

bcw<-read.table("./data/breast-cancer-wisconsin.data",sep=',')

1.2

#Assign names to 11 different columns in my dataset.

names(bcw) <- c("Sample.code.number","Clump.Thickness","Uniformity.of.Cell.Size",

"Uniformity.of.Cell.Shape","Marginal.Adhesion","Single.Epithelial.Cell.Size",

"Bare.Nuclei","Bland.Chromatin","Normal.Nucleoli","Mitoses","Class")

1.3

#1.3See Null value

bcw[complete.cases(bcw),]

#Change column “Bare.Nuclei” from factor to integer

bcw$Bare.Nuclei <- as.integer(as.character(bcw$Bare.Nuclei))

#Remove all Null value

bcw1=na.omit(bcw)

1.4

#Remove the first column.

bcw=bcw[,-1]

1.5

#see the type of Class

class(bcw$Class)

#"integer"

#Change type integer to factor.

bcw[,'Class']<-factor(bcw[,'Class'])

#see the type of Class after changing

class(bcw$Class)

#factor

1.6

#save as bcw\_processed.Rda

saveRDS(bcw1, file="./data/bcw\_processed.Rda")

Save(bcw1, file=”./code/myPreparation.r”)

Task2

2.1

#Load bcw\_processed.Rda

bcw2<-load("bcw\_processed.Rda")

#Select first nine variables

bcw3 <- bcw1[,1:9]

#For reproducible result

set.seed(2835)

2.2

#Cluster the data into 2 clusters

nclust = 2

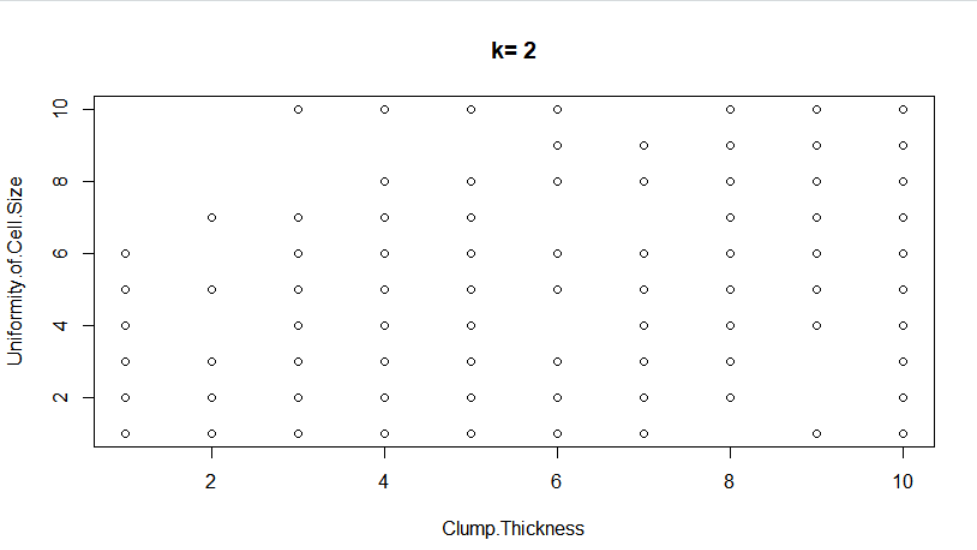
bcw3 <- bcw1[,1:9]

(kmeans.result <- kmeans(bcw3,nclust))

#Plot

plot(bcw1[,c("Clump.Thickness","Uniformity.of.Cell.Size")])

title(paste("k= ",nclust,sep=""))



2.3

#Color the points according to the Class column

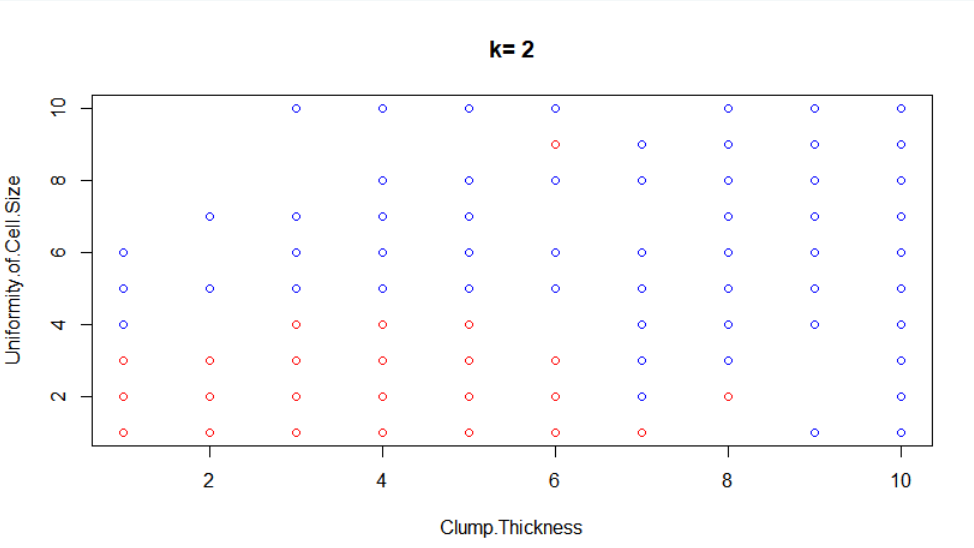
bcw3 <- bcw[,1:9]

nclust = 2

(kmeans.result <- kmeans(bcw3,nclust))

plot(bcw1[,c("Clump.Thickness","Uniformity.of.Cell.Size")],

col = bcw1$Class)

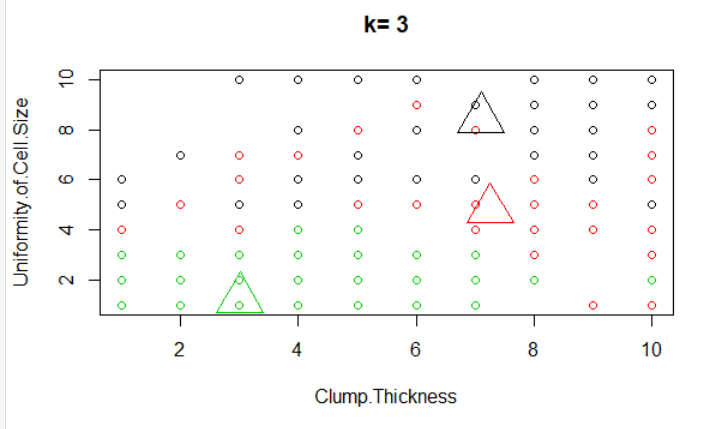
title(paste("k=",nclust,sep=""))

2.4

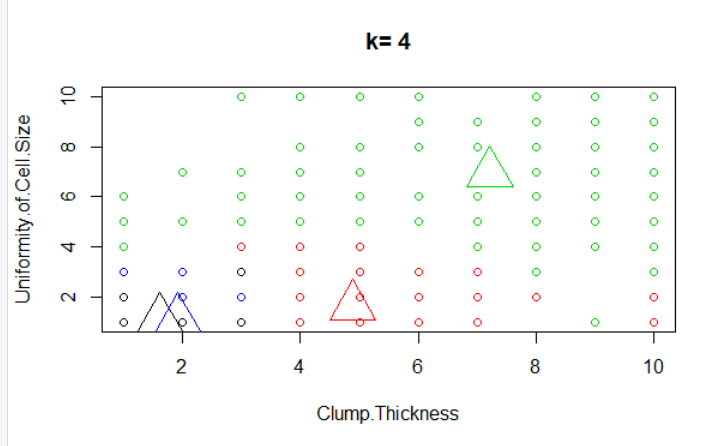
Yes. 2.3 has two colors, which is more obvious. However, before coloring, the graph only has one color and we do not know how to distinguish them.

2.5 #Cluster more

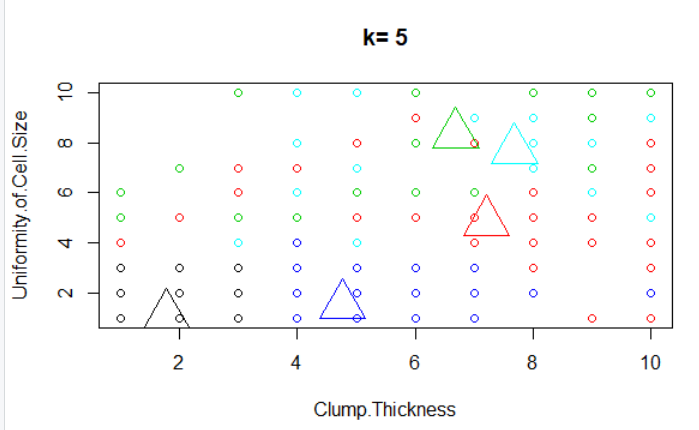
When k=3

****

When k=4



When k=5

****

2.6

According to the plot with k=2,3,4,5. In my opinion, it is not a good choice if we set too many clusters. As k=5, the centroid is very close between two clusters and it looks a little messy, which is not necessary. On the other hand, when there are two or three clusters, it is convenient for us to observe.

2.7 hierarchical clustering

#hierarchical clustering with hclust function

hc <- hclust(dist(bcw1))

#plot the obtained dendrogram

plot(hc,hang=-1)

#Cluster the dendrogram into nclust clusters

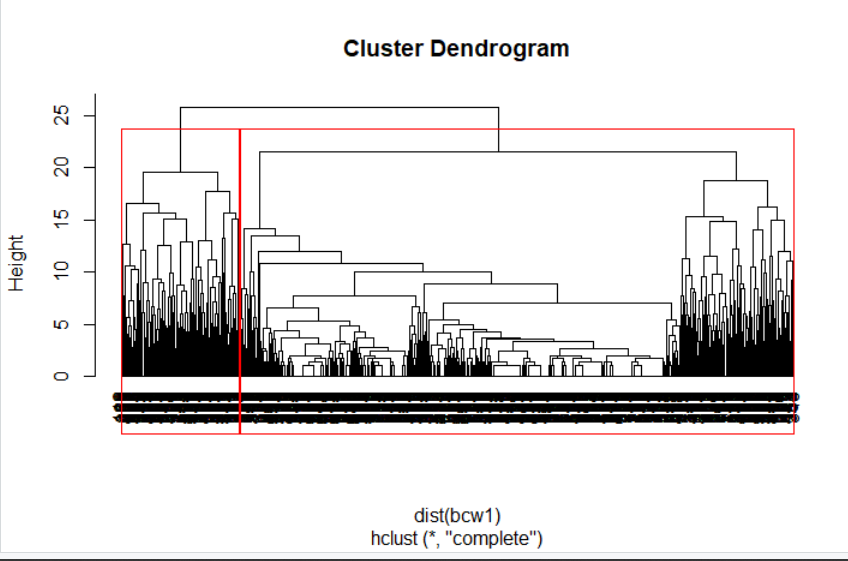
#when n =2

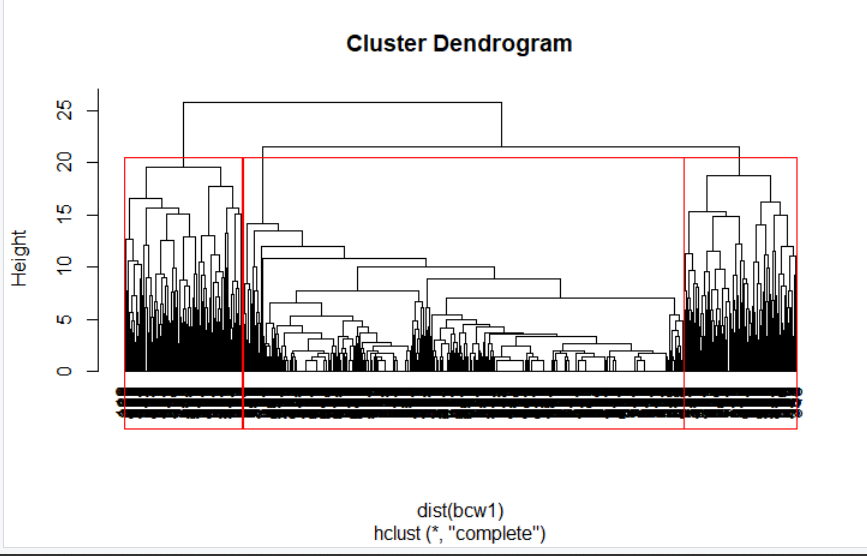
nclust = 2

rect.hclust(hc,k=nclust)

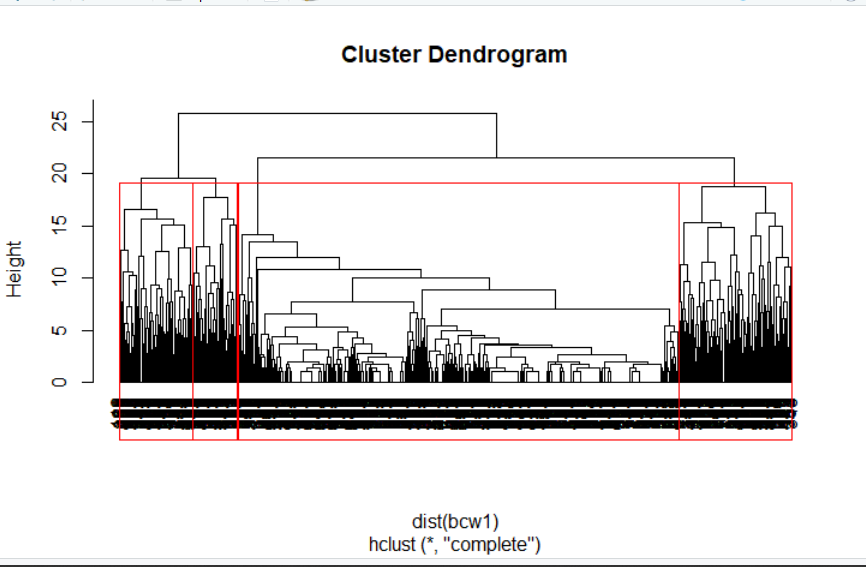
groups <- cutree(hc,k=nclust)

When n=2

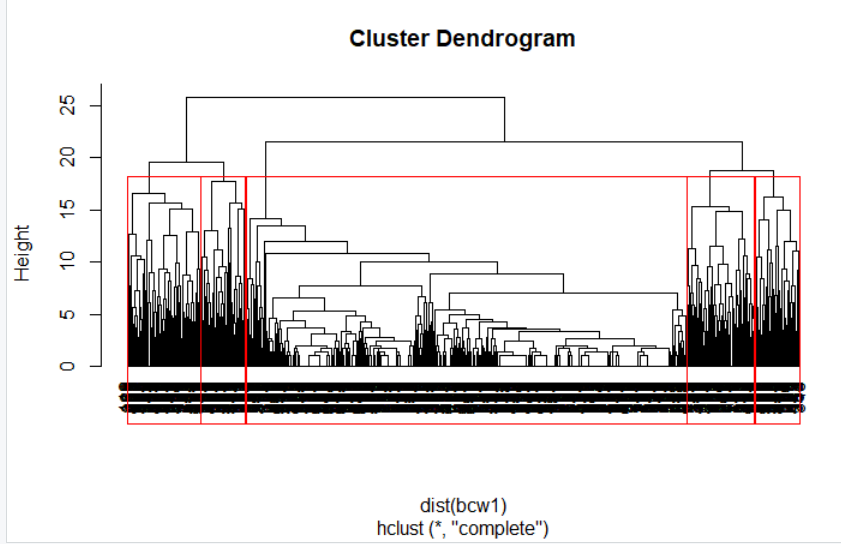
****

n=3****

n =4

****

n=5

****

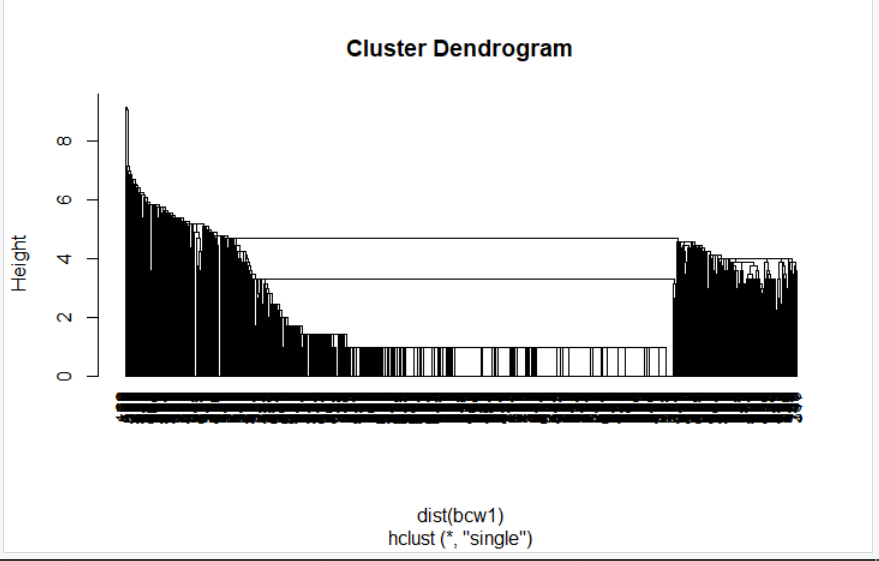
2.8

Yes, because the cluster in the middle is too large.

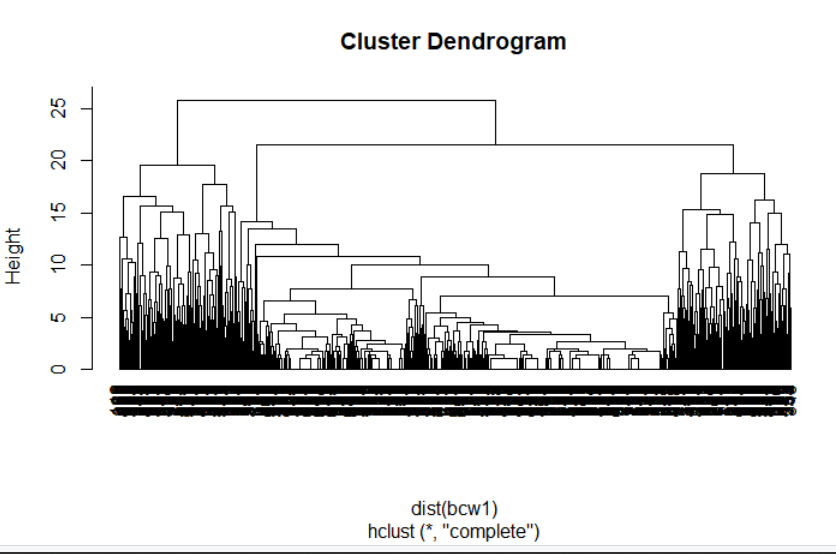
2.9 #Cluster with methods

hc <- hclust(dist(bcw1),method="single")

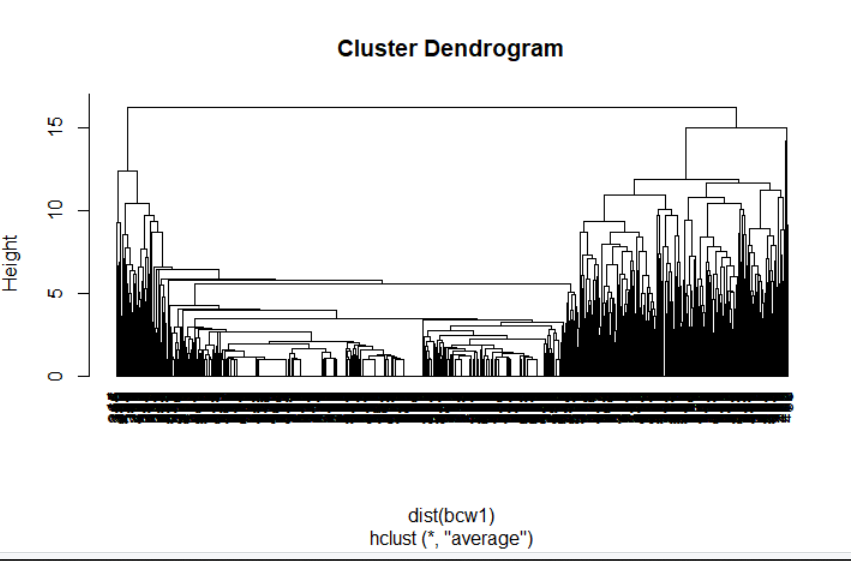
plot(hc,hang=-1)

****

“complete”

****

“average”

****Agglomerative function makes clustering work in a bottom-up manner. After using this function, the hierarchical clustering becomes more obvious and easy to help us to find difference.

Task3

3.1

#3Load bcw\_processed.Rda

bcwtask3<-load("bcw\_processed.Rda")

#Divde dataset into "training" and "test"

set.seed(2835)

m = nrow(bcwtask3)

training\_percentage = 0.7

test\_percentage = 0.3

#Sample random index

ind <- sample(2,m,replace = TRUE, prob = c(training\_percentage,test\_percentage))

#Select training and test data

training\_data = bcw1[ind == 1,]

test\_data = bcw1[ind == 2,]

#divide features and labels

training\_features <- training\_data[,1:9]

training\_labels <- training\_data[,10]

test\_features <- test\_data[,1:9]

test\_labels <- test\_data[,10]

3.2

# install and import "party" library

install.packages("party")

library(party)

#Specify target(class) and predictors(features)

myFormula <- Class ~ Clump.Thickness + Uniformity.of.Cell.Size + Uniformity.of.Cell.Shape +

Marginal.Adhesion + Single.Epithelial.Cell.Size +

Bare.Nuclei + Bland.Chromatin + Normal.Nucleoli + Mitoses

#generate classification tree

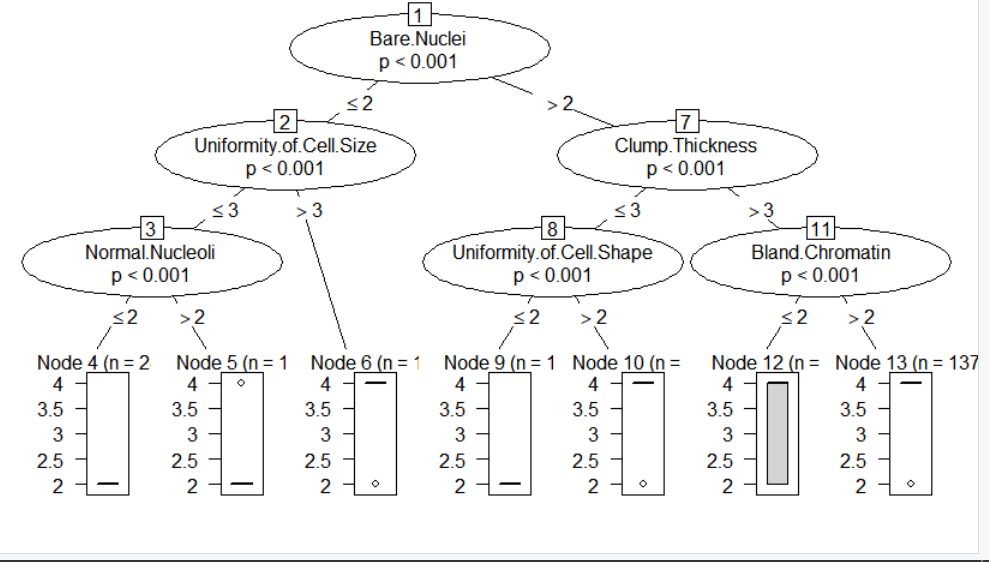
bcwtask3\_ctree <- ctree(myFormula,data=training\_data)

#visualise the tree

plot(bcwtask3\_ctree)

#predict test labels

ctree\_pred<- predict(bcwtask3\_ctree, training\_data)

****

When we have more than two variables, it is better to use classification tree and it can help us to predict. According to output graph, every middle node needs to depend on corresponding variables and the conditions of splitting has shown on the branches. Leaf nodes can show the number of different sample.

3.4 K-MN

#install and import "class" library

install.packages("class")

library(class)

#classify using K-MN

knn\_pred <- knn(train = training\_features,

test = test\_features,

cl = training\_labels,

k=1)

# create the confusion matrix

cm = as.matrix(table(Actual = test\_labels, Predicted = ctree\_pred))

n = sum(cm) #number of instances

nc = nrow(cm) #number of classes

diag = diag(cm) #number of correctly classified instances per class

rowsums = apply(cm,1,sum)#number of instances per class

colsums = apply(cm,2,sum)#number of predictions per class

#compute accruacy, precision, recall and f1

accruacy = sum(diag) / n

precision = diag / colsums

recall = diag / rowsums

f1 = 2 \* precision \* recall/(precsion + recall)

results <- data.frame(precision,recall,f1)