Lab 6 Cluster Analysis Part II

The goal of this lab is to become familiar with the application of **Polythetic Hierarchical Divisive** and **Agglomerative Clustering**.

# Set up R session

## Data

Call in the data set “Caribbean\_birds.csv” from your working directory and call it ‘birds’. This data set consists of presence-absence data for bird species on 19 Caribbean Islands.

birds<-read.csv("../Data/lab\_6/Caribbean\_birds.csv", row=1, header=TRUE)

## Download packages

We will be using the following packages:

library(raster)  
library(cluster)  
library(vegan)  
library(pvclust)

# Calculating the distance/dissimilarity matrix

Select the (an) appropriate dissimilarity metric for binary data and calculate the dissimilarity/distance matrix.

distBirds<-vegdist(birds,"jaccard")

# Polythetic Agglomerative Hierarchical Clustering (PAHC)

You will use the hclust function in the *stats* package to conduct PAHC. This hclust function contains the six fusion methods we discussed in lecture. We will use hclust to cluster the Caribbean bird data and construct dendrograms.

?hclust

## Clustering algorithms

singleTree<-hclust(distBirds, method = "single")  
completeTree<-hclust(distBirds, method = "complete")  
centroidTree<-hclust(distBirds, method = "centroid")  
medianTree<-hclust(distBirds, method = "median")  
averageTree<-hclust(distBirds, method = "average")  
wardTree<-hclust(distBirds, method = "ward.D2")

Let’s plot each dendrogram individual and explore the patterns:

plot(singleTree)  
plot(completeTree)  
plot(centroidTree)  
plot(medianTree)  
plot(averageTree)  
plot(wardTree)

Let’s now look at the dendrograms on the same plot:

par(mfrow=c(2,3))  
plot(singleTree)  
plot(completeTree)  
plot(centroidTree)  
plot(medianTree)  
plot(averageTree)  
plot(wardTree)

What groups are clustering together? Do the clusters change with different methods?

## Evaluating the cluster solution

We learned in lecture three main ways to evaluate the cluster solution. The *agglomerative coefficient*, the *cophenetic correlation coefficient*, and *Monte Carlo* simulations (i.e. bootstrapping).

### Agglomerative coefficient

First, calculate the *agglomerative coefficient* for each fusion method:

ag1<-coef.hclust(singleTree)  
ag2<-coef.hclust(completeTree)  
ag3<-NA  
ag4<-NA  
ag5<-coef.hclust(averageTree)  
ag6<-coef.hclust(wardTree)

Now lets put them in a table:

methods<-c("single","complete","centroid", "median", "average", "ward")  
agc<-round(c(ag1,ag2,ag3,ag4,ag5,ag6),2)  
agcTable<-data.frame(methods,agc)

### Cophenetic correlation coefficient

Next, let’s calculate the *cophenetic correlation coefficient*. This will allow us to see how well the dendrogram built by each fusion method reproduce the original distance matrix and will also allow us to compare the different fusion methods:

cc1<-cor(distBirds,cophenetic(singleTree))  
cc2<-cor(distBirds,cophenetic(completeTree))  
cc3<-cor(distBirds,cophenetic(centroidTree))  
cc4<-cor(distBirds,cophenetic(medianTree))  
cc5<-cor(distBirds,cophenetic(averageTree))  
cc6<-cor(distBirds,cophenetic(wardTree))  
cophCor<-round(c(cc1,cc2,cc3,cc4,cc5,cc6),2)

Let’s put this all in a table:

methods<-c("single","complete","centroid", "median", "average", "ward")  
dendrogramTable<-data.frame(methods,cophCor,agc)  
dendrogramTable

## methods cophCor agc  
## 1 single 0.95 0.51  
## 2 complete 0.95 0.61  
## 3 centroid 0.86 NA  
## 4 median 0.75 NA  
## 5 average 0.96 0.57  
## 6 ward 0.91 0.79

**A loop!**

This loop can be applied above anytime you are doing the same function (e.g., hclust) but one thing is changing (i.e., fusion method). Here, I show it for calculating the cophenetic correlation. What we are doing is looping the function for calculating the cophenetic correlation over a list of names that refer to dendrograms made with different linkage methods.

#function telling the loop to read the input as text  
e = function(expr) eval(parse(text=expr))   
  
#sets up a variable to fill with the output of the loop  
cc<-NULL   
  
#list of names for the loop  
methodList <- c("singleTree", "completeTree", "centroidTree", "medianTree", "averageTree", "wardTree")   
  
# run the loop  
for (i in methodList) {  
 cc[i]<-round(cor(distBirds,cophenetic(e(i))),2)  
}  
  
cc

## singleTree completeTree centroidTree medianTree averageTree wardTree   
## 0.95 0.95 0.86 0.75 0.96 0.91

### Bootstrapping

Last but not least, let’s run a bootstrap permutation to see how many clusters are “good clusters”. We are going to use the function pvclust in the *pvclust* package. Since we know that “single”, “complete”, and “average” linkage methods had the highest cophenetic correlation, let’s focus on these:

?pvclust

pvclust only has functionality for certain types od dissimilarity/distance types “method.dist=”. Often, we will use a different measure and need to define custom distance function which returns an object of class “dist”. Here we do it for jaccard.

jaccard <- function(x) {  
 x <- t(as.matrix(x))  
 res <- vegdist(x, method="jaccard")  
 res <- as.dist(res)  
 attr(res, "method") <- "jaccard"  
 return(res)  
}

The resampling procedure takes a little bit of time so we will only use 100 bootstraps (nboot = 100). Normally, you would conduct at least 1000:

boot1<-pvclust(t(birds), method.hclust="single",method.dist=jaccard,iseed=22, nboot=100)  
boot2<-pvclust(t(birds), method.hclust="complete",method.dist=jaccard,iseed=22, nboot=100)  
boot3<-pvclust(t(birds), method.hclust="average",method.dist=jaccard,iseed=22, nboot=100)  
  
#Here, jaccard is our custom distance/dissimilarity function

Now plot each dendrogram with the p-values from the Monte Carlo simulation. The “au” p-values (in red) correspond to multi-scale bootstrapping, while the “bp” p-values correspond to normal bootstrapping. The first row of plots has rectangles (function pvrect) around the largest cluster with p-values ≤ 0.05 according to the “au” method. The second row of plots has rectangles placed around the largest clusters with p-values ≤ 0.05 according to the “bp” method.

?pvrect  
  
par(mfrow=c(2,3))  
  
plot(boot1)  
pvrect(boot1, alpha=0.95, pv="au")  
plot(boot2)  
pvrect(boot2, alpha=0.95, pv="au")  
plot(boot3)  
pvrect(boot3, alpha=0.95, pv="au")  
  
plot(boot1)  
pvrect(boot1, alpha=0.95, pv="bp")  
plot(boot2)  
pvrect(boot2, alpha=0.95, pv="bp")  
plot(boot3)  
pvrect(boot3, alpha=0.95, pv="bp")

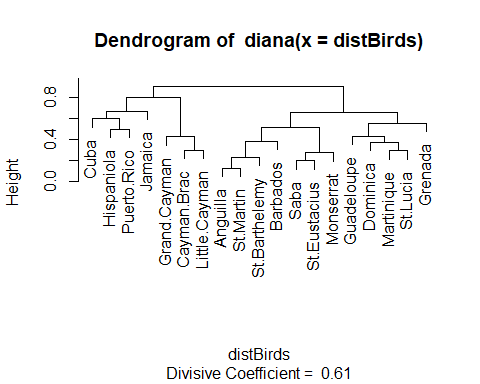
# Polythetic Divisive Hierarchical Clustering (PDHC)

You will use the diana function in the *cluster* package to conduct PDHC. We will use diana to cluster the Caribbean bird data and construct dendrograms.

?diana

## Clustering algorithm

diTree <- diana(distBirds)  
  
#Next, plot the dendrogram:  
  
plot(diTree, which.plots = 2)



#calculate the divisive coefficient  
  
diTree$dc

## [1] 0.6052537

#and calculate the cophenetic correlation coefficient:  
  
d.coph <- cor(distBirds,cophenetic(diTree))

**Now use these hierarchical methods to cluster your own data. If you don’t have your own data and are brave, try the island birds of the Atlantic Ocean (atlantic\_birds.csv).**