Lab 6 Cluster Analysis Part II

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

## 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("G:/Shared drives/MultivariateStatistics/Data/LabData/Lab6/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

For this lab, we will use Jaccard’s dissimilarity metric for calculating the dissimilarity matrix.

## Question 1: Why is Jaccard’s an appropriate dissimilarity metric for this dataset? (10 pts)

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

Note that the method argument in ‘hclust’ specifies the fusion method.

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 individually 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. Note that this won’t print nicely in your Word document, but you’ll be able to see it if you make the upper left editor tab in RStudio large:

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

## Question 2: What groups are clustering together? Do the clusters change with different methods? (20 pts)

## 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)

## Question 3: Why is the agglomerative coeffient ‘NA’ for the centroid and median fusion methods? (10 pts)

### 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 reproduces 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 to show you another way to generate the values of the table we just made. 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. note that growing  
# objects from a for loop is not a good idea for more computationally onerous  
# calculations, but it's ok here.  
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 statistically significant. 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)

## starting httpd help server ... done

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 = "binary", nboot = 100)  
boot2 <- pvclust(t(birds), method.hclust = "complete", method.dist = "binary", nboot = 100)  
boot3 <- pvclust(t(birds), method.hclust = "average", method.dist = "binary", nboot = 100)  
  
# Here, 'binary' is Jaccard

## Question 4: Why is it preferable to increase the number of bootstrap iterations when possible computationally? (10 pts)

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 deemed significant at an alpha level of 0.05 according to the “au” method. The second row of plots has rectangles placed around the largest clusters deemed significant at an alpha level of 0.05 according to the “bp” method. Note that rather than listing significant p-values as those <0.05, this function lists them as >0.95, which is a bit confusing. This is described on the Github [repository](https://github.com/shimo-lab/pvclust) for the package under the *Instruction* section: “For a cluster with AU p-value > 0.95, the hypothesis that”the cluster does not exist” is rejected with significance level 0.05; roughly speaking, we can think that these highlighted clusters does not only “seem to exist” caused by sampling error, but may stably be observed if we increase the number of observation.”

`?`(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")

## Question 5: Would PAHC be appropriate for using on your data or a subset of your data? (50 pts)

***Optional:*** **If you have extra time, try the these analyses on the island birds of the Atlantic Ocean (atlantic\_birds.csv).**