

Laboratorio 7: Clasificación Jerárquica

Fecha de entrega: ver Mediación Virtual

Instrucciones: Resuelva los ejercicios propuestos en el Laboratorio. Guarde el script respectivo, con su nombre y súbalo en el sitio del curso. Además, guarde los resultados y súbalos también en el sitio.

1. Cargar la tabla de datos de los Iris de Fisher:

```
data(iris)
names(iris)
X<-na.omit(iris[,1:4])
X
```

2. Calcular las distancias Euclídeas: `d <- dist(X, method = 'euclidean')`
Otras distancias son: "maximum", "manhattan", "canberra", "binary" o "minkowski"
La función `vegdist()` tiene más disimilitudes (por defecto usa Bray & Curtis):

```
"manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard",
"gower", "altGower", "morisita", "horn", "mountford", "raup", "binomial" o
"chao".
```

3. **Clasificación Jerárquica Ascendente:** la función predefinida en R es `hclust`. Necesita una matriz de disimilitudes y un método (criterio de agregación).

```
clasif <- hclust(d, method = "complete")
```

Los otros criterios de agregación ("method") son:

```
"ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA),
"median" (= WPGMC), "centroid" (= UPGMC).
```

(los dos últimos pueden presentar inversiones). El criterio clásico de Ward se obtiene con "ward.D2" en el que las distancias están al cuadrado, tal como se presenta en el libro.

4. Dibujo del árbol o dendrograma: `plot(clasif)`.

Parámetros importantes:

- "labels": TRUE si se quiere que aparezcan las etiquetas de los objetos a clasificar.
- **Corte del árbol:** `corte <- cutree(clasif, k = 3)`
- Para ver el corte de las clases sobre el árbol:

```
plot(clasif, labels= FALSE)
rect.hclust(clasif, k=3, border = "red")
```

5. Se pueden ver diferentes árboles, con el mismo criterio de agregación, pero con distinta distancia:

```
d.L2 <- dist(X, method = 'euclidean')
d.L1 <- dist(X, method = 'manhattan')
d.max <- dist(X, method = 'maximum')
d.canb <- dist(X, method = 'canberra')
```

```
clasif <- hclust(d.L2, method = "average")
```

```

plot(clasif, labels= FALSE)
rect.hclust(clasif, k=3, border = "red")
clasif <- hclust(d.L1, method = "average")
plot(clasif, labels= FALSE)
rect.hclust(clasif, k=3, border = "red")
clasif <- hclust(d.max, method = "average")
plot(clasif, labels= FALSE)
rect.hclust(clasif, k=3, border = "red")
clasif <- hclust(d.canb, method = "average")
plot(clasif, labels= FALSE)
rect.hclust(clasif, k=3, border = "red")

```

6. Ejemplo de los carros:

```

data(mtcars)
names(mtcars)
mtcars
clasif.carros <- hclust(dist(mtcars), method = "average")
plot(clasif.carros)
corte <- cutree(clasif.carros, k = 3)
corte
rect.hclust(clasif.carros, k=3, border = "red")

```

7. El objetivo de esta parte es ver distintas variantes para hacer **dendrogramas**.

- (a) Correr `clasif= hclust(dist(mtcars))`. El dendrograma más sencillo se obtiene con `plot(clasif)`
Véase que las etiquetas de los objetos están a distintas alturas.
- (b) Para desplegar las etiquetas en el mismo nivel, hacer: `plot(clasif, hang = -1)`
- (c) Crear un objeto tipo dendrograma:

```

arbol = as.dendrogram(clasif)
plot(arbol)

```

- (d) El árbol se puede graficar en forma triangular:

```

plot(arbol, type = "triangle")

```

- (e) El dendrograma se puede cortar en la parte superior:

```

op = par(mfrow = c(2, 1))
plot(cut(arbol, h = 75)$upper, main = "Parte superior cortando en h=75")
plot(cut(arbol, h = 75)$lower[[2]], main = "Segunda rama de arbol inferior cortando en h=75")

```

- (f) Gráfico sofisticado, con colores:

- Primero se crea un vector de colores:
`labelColors = c("#CDB380", "#036564", "#EB6841", "#EDC951")`
- Para cortar el dendrograma en 4 clases
`clusMember = cutree(clasif, 4)`
- Función para obtener las etiquetas de los colores:

```

collab <- function(n) {
  if (is.leaf(n)) {
    a <- attributes(n)
    labCol <- labelColors[clusMember[which(names(clusMember) == a$label)]]
    attr(n, "nodePar") <- c(a$nodePar, lab.col = labCol)
  }
  n
}

```

- Usar `dendrapply`:
`clusDendro = dendrapply(arbol, colLab)`
 - Graficar:
`plot(clusDendro, main = "Dendrograma Bonito", type = "triangle")`
- (g) Manejo de algunos parámetros:
- ```
op = par(bg = "#DDE3CA")
plot(clasif, col = "#487AA1", col.main = "#45ADA8", col.lab = "#7C8071",
 col.axis = "#F38630", lwd = 3, lty = 3, sub = "", hang = -1, axes = FALSE)
```
- (h) Añadir ejes:
- ```
axis(side = 2, at = seq(0, 400, 100), col = "#F38630", labels = FALSE,
      lwd = 2)
```
- (i) Añadir texto al margen:
- ```
mtext(seq(0, 400, 100), side = 2, at = seq(0, 400, 100), line = 1,
 col = "#A38630", las = 2)
```
- (j) Para convertir a tipo de árbol filogenético horizontal, hacer `install.packages('ape')` y `library(ape)`
- (k) Gráfico básico horizontal:
- ```
plot(as.phylo(clasif), cex = 0.9, label.offset = 1)
```
- (l) Otros gráficos de `plot.phylo`:
- cladograma:
`plot(as.phylo(clasif), type = "cladogram", cex = 0.9, label.offset = 1)`
 - sin raíz:
`plot(as.phylo(clasif), type = "unrooted")`
 - fan (aspas):
`plot(as.phylo(clasif), type = "fan")`
 - radial:
`plot(as.phylo(clasif), type = "radial")`
- (m) Árboles filogenéticos con algunas variantes.
- Añadir colores al azar:
`plot(as.phylo(clasif), type = "fan", tip.color = hsv(runif(15, 0.65, 0.95), 1, 1, 0.7), edge.color = hsv(runif(10, 0.65, 0.75), 1, 1, 0.7), edge.width = runif(20, 0.5, 3), use.edge.length = TRUE, col = "gray80")`
 - Definir vector de colores, cortar el dendrograma en 5 clases y hacer que el tamaño refleje las millas por galón:
`mypal = c("#556270", "#4ECDC4", "#1B676B", "#FF6B6B", "#C44D58")`
`clus5 = cutree(clasif, 5)`
`op = par(bg = "#E8DDCB")`
`plot(as.phylo(clasif), type = "fan", tip.color = mypal[clus5], label.offset = 1, cex = log(mtcars$mpg, 10), col = "red")`
- (n) Gráficos con diversas variantes manejando colores, fondos y fuentes:
- Paquete `sparcl`:
 - Instalar el paquete: `install.packages('sparcl')`
 - Cargar `library(sparcl)`
 - Colores de las hojas del árbol: `y = cutree(clasif, 3)`
 - Colorear:

```

    ColorDendrogram(clasif, y = y, labels = names(y), main = "Datos de Carros",
                     branchlength = 80)
ii. Paquete ggdendro
    • install.packages('ggdendro')
    • library(ggdendro)
    • ggdendrogram(clasif)
    • Rotando
      ggdendrogram(clasif, rotate = TRUE, size = 4, theme_dendro = FALSE, color = "red")
iii. Líneas triangulares con el paquete ggplot2
    • install.packages('ggplot2')
    • library(ggplot2)
    • ddata <- dendro_data(as.dendrogram(clasif), type = "triangle")
      ggplot(segment(ddata)) + geom_segment(aes(x = x, y = y, xend = xend,
        yend = yend)) + ylim(-10, 150) + geom_text(data = label(ddata), aes(x = x,
        y = y, label = label), angle = 90, lineheight = 0)
iv. Dendrogramas de Romain François, "Addicted to R".
    • Cargar función de A2R
      source("http://addictedtor.free.fr/packages/A2R/lastVersion/R/code.R")
    • Dendrograma coloreado
      op = par(bg = "#EFEFEF")
      A2Rplot(clasif, k = 3, boxes = FALSE, col.up = "gray50", col.down = c("#FF6B6B",
        "#4ECDC4", "#556270"))
      par(op)
    • Otro dendrograma coloreado con fondo negro:
      op = par(bg = "gray15")
      cols = hsv(c(0.2, 0.57, 0.95), 1, 1, 0.8)
      A2Rplot(clasif, k = 3, boxes = FALSE, col.up = "gray50", col.down = cols)

```