UNIVERSITAT DE BARCELONA

Universitat Oberta de Catalunya

Análisis de Conglomerados

Francesc Carmona

12 de mayo de 2019

1. Tenemos 5 objetos A, B, C, D, E. Dibujar tres dendogramas en los casos que (a) sólo hay un conglomerado o *cluster*, (b) los conglomerados son $\{A, B\} \subset \{A, B, C\} \subset \{A, B, C, D\} \subset \{A, B, C, D, E\}$, (c) los conglomerados son $\{A, B\} \subset \{A, B, C\}, \{D, E\}$.

   Inventar una distancia entre los objetos de forma que la clasificación jerárquica proporcione los conglomerados del caso (c). Probar que es correcta con la función `hclust()` y dibujar el dendograma con la ayuda de la función `plot()` de **R**.

2. El *data.frame* `all.mammals.milk.1956` del paquete `cluster.datasets` contiene los datos de los ingredientes en la leche materna de 25 animales[1].

   Las variables medidas son agua, proteínas, grasa, lactosa y cenizas (los valores están en porcentaje).

   a) Realizar un análisis de proximidades o escalado multidimensional con la distancia euclídea.

   b) Realizar un análisis de conglomerados jerárquico con las distancias euclídeas y el método de Ward.

   c) (∗) Repetir los dos análisis anteriores con la distancia de Bhattacharyya.

   Nota: Entre los criterios para calcular la distancia entre la unión de dos conglomerados y los otros, Everitt(2005) comenta los tres más conocidos: método del mínimo (single linkage), método del máximo (complete linkage) y método de la media (average linkage). Cuando los datos no son métricos, la media se puede reemplazar con la mediana y el método es *median linkage*. Además, existen otros como el método del centroide y el de Ward. La función `hclust()` dispone de todos ellos.

   El método del centroide consiste en hallar un objeto "medio" en cada conglomerado y calcular la distancia entre conglomerados como la distancia entre sus objetos medios.

   El método de Ward, también llamado el método de la varianza mínima, consiste en una nueva estrategia en el primer paso del algoritmo. En lugar de unir los dos conglomerados más próximos, el método de Ward busca la unión de los dos conglomerados cuya mezcla consiga la menor suma de cuadrados dentro del conglomerado (minimum within-group variance).

   Actualmente hay dos algoritmos distintos para el procedimiento de Ward. El que se ejecuta con la opción `"ward.D"` (equivalente a la opción `"ward"` en versiones de **R** <= 3.0.3) no implementa el criterio de conglomerados de Ward (1963), mientras que la opción `"ward.D2"` sí lo hace. La función `agnes(*, method="ward")` corresponde a `hclust(*, "ward.D2")`.

   En el ejercicio 6(f) se hace una comparativa de los dos algoritmos con estos datos.

3. El paquete `cluster` de **R** dispone de los datos Ruspini, muy conocidos en la literatura del análisis de conglomerados.

```
library(cluster)
data(ruspini)
help(ruspini)
str(ruspini)
```

[1] También se pueden obtener en el archivo `milk` del paquete `flexclust` de **R**.

a) Aplicar el procedimiento de particionado de las k-medias con el algoritmo de Hartigan-Wong[2] para 4 grupos.

Dibujar los puntos con un símbolo distinto para cada grupo. Hallar los centros de cada grupo y dibujarlos en el gráfico anterior. Hallar las sumas de cuadrados dentro de cada grupo y los tamaños de los conglomerados.

b) Realizar un particionado alrededor de los medoides[3] o k-medoides o PAM con la función `pam()` del paquete cluster de **R** y hallar los medoides.

c) Calcular la silueta[4] como medida de la calidad de los conglomerados formados.

La función `silhouette()` nos puede servir. Realizar un `summary()` y un `plot()` sobre el objeto silueta.

d) Realizar un particionado jerárquico con la función `agnes()` del paquete cluster. El código es

```
ar <- agnes(ruspini)
cluster4 <- cutree(ar, k=4)
si4 <- silhouette(cluster4, daisy(ruspini))
plot(si4)
```

Probar con diversos números de conglomerados y decidir la mejor partición.

Una buena idea es representar los conglomerados resultantes. La función

```
clusplot(ruspini, cluster4, color=TRUE, shade=TRUE,
         labels=2, lines=0)
```

del paqueste cluster con unos cuantos parámetros nos puede ayudar en la visualización gráfica del resultado.

4. Realizar un particionado alrededor de los k-medoides con los datos de la leche materna del ejercicio 2.

   Estudiar el mejor número de conglomerados con ayuda de sus siluetas.

   Dibujar un gráfico bidimensional con los conglomerados hallados.

5. En el *data.frame* `birth.death.rates.1966` del paquete `cluster.datasets` tenemos los datos en porcentaje de nacimientos y muertes de 70 países[5].

   a) Realizar de forma exploratoria un análisis de conglomerados jerárquico con la función `agnes()` y su dendograma.

   b) Por el apartado anterior parece que hay tres grupos. Realizar un PAM con $k = 3$ grupos y probar distintos números con ayuda de sus siluetas.

   c) Comparar el resultado del método de los k-medoides con el de las k-medias. Realizar una tabla cruzada de clasificación.

6. ($*$) El paquete `dendextend` proporciona un conjunto de funciones para mejorar gráficamente los dendogramas. Con la clasificación jerárquica en cuatro grupos obtenida por el método de Ward en el ejercicio 2 con los datos de la leche materna de 25 animales, realizar el siguiente análisis gráfico:

   a) Empezaremos con un gráfico del tipo matriz de diagramas de dispersión (*scatterplot matrix* o SPLOM) que podemos implementar con la función `pairs()` con los puntos en colores distintos según el conglomerado.

   b) Como tenemos pocas variables, también se puede hacer un gráfico de coordenadas paralelas con la ayuda de la función `parcoord()` del paquete `MASS`.

[2] La función `kmeans` puede ser útil.

[3] `http://en.wikipedia.org/wiki/K-medoids`

[4] `http://en.wikipedia.org/wiki/Silhouette_(clustering)`

[5] También se pueden obtener en el archivo `birth` del paquete flexclust de **R**.

c) Dibujar el dendograma en horizontal con cuatro colores, uno por conglomerado, tanto para las ramas como para las etiquetas.

d) Repetir el dendograma en forma circular con la función `circlize_dendogram()` que requiere que el paquete `circlize` esté instalado.

e) Realizar un *heatmap* del dendograma con la función `heatmap.2()` del paquete `gplots`.

f) Comparar las clasificaciones que proporcionan los algoritmos `"ward.D"` y `"ward.D2"` con la función `tanglegram()`.

Nota: Además de consultar la ayuda y la viñeta del paquete `dendextend`, también se puede aprender mucho con los tres ejemplos de la página

`https://cran.r-project.org/web/packages/dendextend/vignettes/Cluster_Analysis.html`