

Mineria de Datos

Clustering II

Dr. Edgar Acuna
Departamento de Matematicas

Universidad de Puerto Rico- Mayaguez

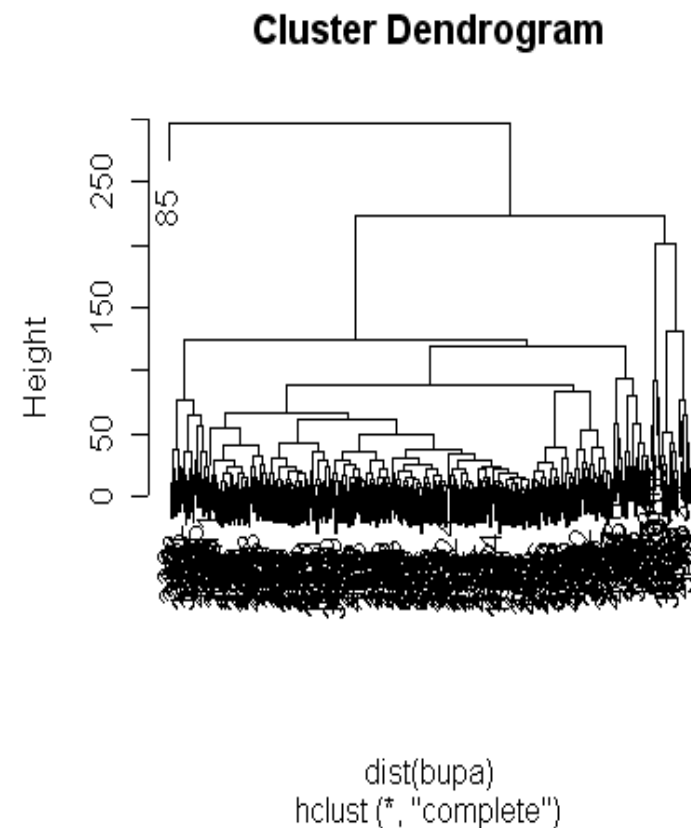
math.uprrm.edu/~edgar

Algoritmos jerárquicos

Estos algoritmos generan sucesiones anidadas de clusters que se pueden visualizar con una estructura de árbol llamado **Dendrograma**,

En *la figura* se muestra el dendrograma del conjunto Bupa obtenido usando la función **hclust** para algoritmo jerárquico aglomerativo de la librería **stats**.

```
> a=hclust(dist(bupa[,1:6]))  
> plot(a)
```

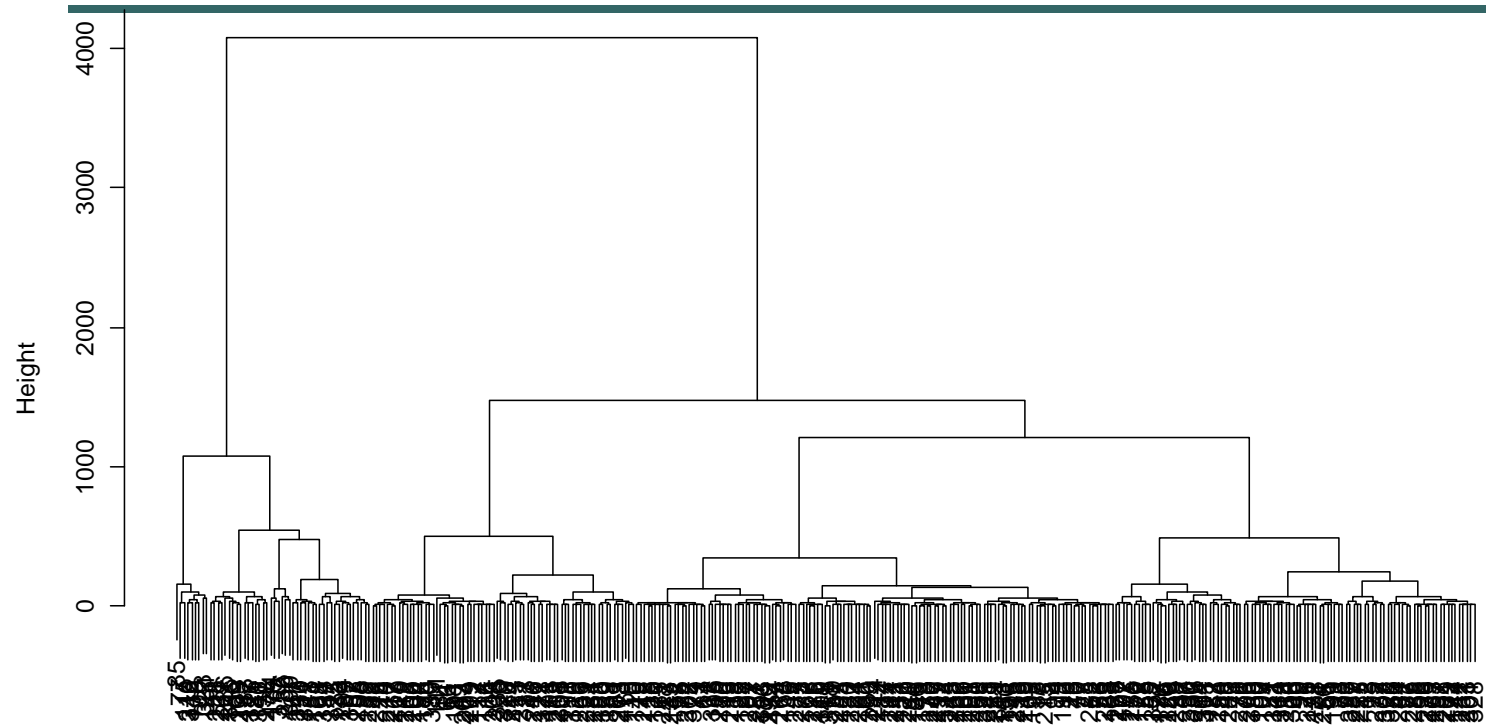


Dendrogramas

Los dendrogramas son fáciles de interpretar pero pueden conducir a falsas conclusiones por las siguientes razones:

- 1) El dendrograma correspondiente a un conglomerado jerárquico no es único, puesto que por cada junte de clusters (merge) uno necesita especificar que sub-árbol va a la derecha y cuál a la izquierda.
Por default la función **hclust** de la librería **stats** ordena los arboles de tal manera que los conglomerados más concentrados van a la izquierda.
- 2) La estructura jerárquica del Dendrograma no representa con certeza las verdaderas distancias entre los objetos distintos del conjunto de datos.

Cluster Dendrogram



dist(bupa[, 1:6])
hclust (*, "ward")

El coeficiente de **correlación cofenético** puede ser usado para medir cuan bien la estructura jerárquica del dendrograma representa a las verdaderas distancias. Se define como la correlación entre las $n(n - 1)/2$ pares de dissimilaridades y sus distancias cofenéticas del dendrograma. La función **cophenetic** en la librería **mva** calcula la distancia cofenéticas.

```
disbupa=dist(bupa[,1:6])
```

```
hbupa=hclust(disbupa, method="ave")
```

```
denbupa=cophenetic(hbupa)
```

```
cor(disbupa,denbupa)
```

La correlacion cofenética da 0.915849

Ejemplo de un dendrograma y sus cortes

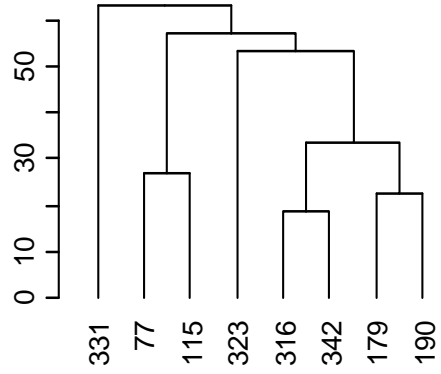
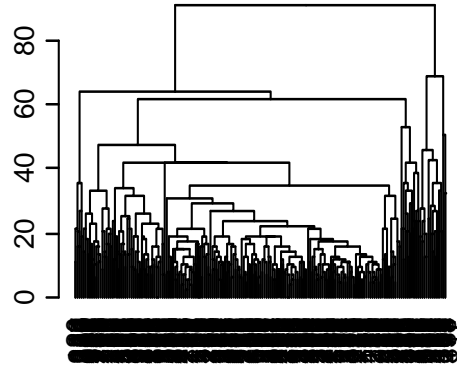
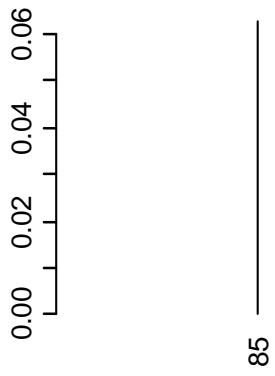
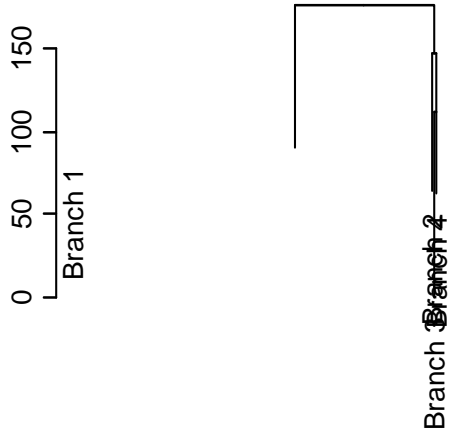
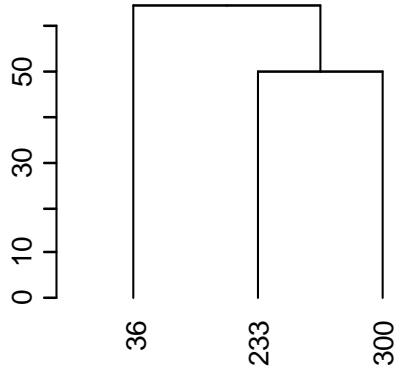
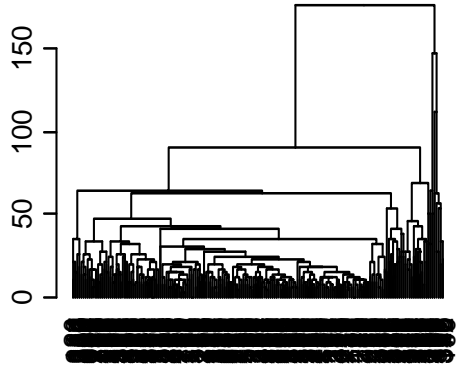
```
treebupa=as.dendrogram(hbupa)
bupita=cut(treebupa, h=100)
> bupita
$upper
`dendrogram' with 2 branches and 4 members total, at height 176.9354
$lower
$lower[[1]]
`dendrogram' with 2 branches and 333 members total, at height 91.22225
$lower[[2]]
`dendrogram' with 2 branches and 3 members total, at height 64.38526
$lower[[3]]
`dendrogram' leaf '85', at height 0
$lower[[4]]
`dendrogram' with 2 branches and 8 members total, at height 62.8725
```

Ejemplo de un dendrograma y sus cortes

```
treebupa=as.dendrogram(hbupa)
bupita=cut(treebupa, h=100)
> bupita
$upper
`dendrogram' with 2 branches and 4 members total, at height 176.9354
$lower
$lower[[1]]
`dendrogram' with 2 branches and 333 members total, at height 91.22225
$lower[[2]]
`dendrogram' with 2 branches and 3 members total, at height 64.38526
$lower[[3]]
`dendrogram' leaf '85', at height 0
$lower[[4]]
`dendrogram' with 2 branches and 8 members total, at height 62.8725
```

Ejemplo de un dendrograma y sus cortes

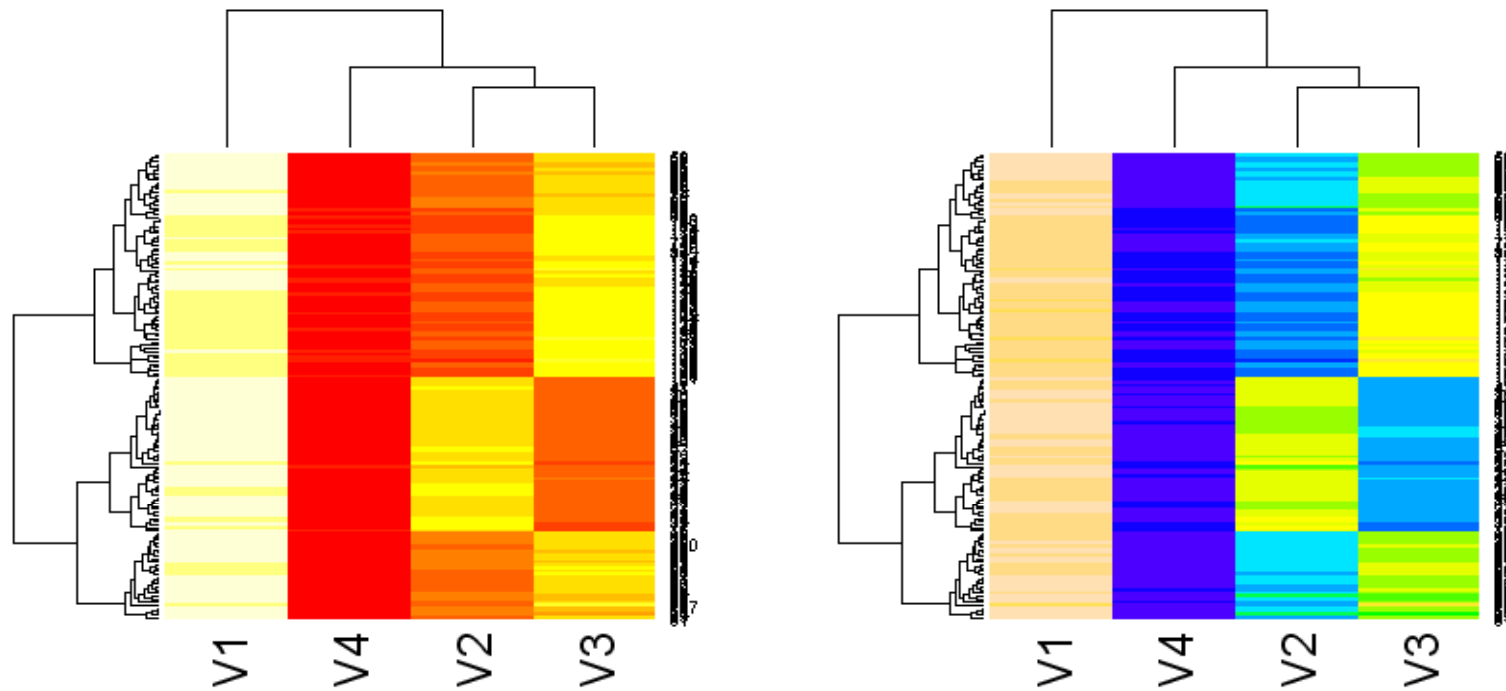
```
> par(mfrow=c(2,3))
> plot(treebupa)
> plot(bupita$upper)
> plot(bupita$lower[[1]])
> plot(bupita$lower[[2]])
> plot(bupita$lower[[3]])
> plot(bupita$lower[[4]])
>
```

Heatmaps.

Son gráficas que muestran simultáneamente las agrupaciones en conglomerados de columna y filas. La función `heatmap` de la librería `mva` permite hacer heatmaps usando un gran número de tonalidades de colores.

Ejemplo de heatmaps para Iris



Notar que solo las variables 2 y 3 determinan claramente las 3 clases.

Algoritmo jerárquico aglomerativo

Suponiendo que tenemos una matriz de datos $m \times n$.

Se empieza con m conglomerados si se desea formar grupos de **muestras** (filas) o con n clusters si se quieren formar grupos de **variables** (columnas).

En cada paso se juntan los cluster mas cercanos usando una medida de **distancia** entre clusters (linkage)

Entre estas distancias están

Linkage promedio: promedio de las distancias de las observaciones en cada cluster.

Linkage simple: la menor distancia entre las observaciones de cada cluster

Linkage completo: la mayor distancia entre las observaciones de cada cluster.

Ejemplo de Jerarquico Aglomerativo

En este caso usaremos la función **agnes** de la librería **cluster**

```
bupagl<-agnes(bupa[,1:6],metric="euclidean",method="ward")
cutree(bupagl,k=2)
```

```
> table(cutree(bupagl,k=2))
```

```
1 2
```

```
62 283
```

```
> table(cutree(bupagl,k=3))
```

```
1 2 3
```

```
283 53 9
```

La función **plot.agnes** permite hacer un **plot del dendrograma**.
Pero no tiene la misma flexibilidad de las funciones en stats.

Métodos jerárquicos divisivos

Empieza con un solo cluster, que es aquel que contiene a todas las muestras.

En cada paso se divide los clusters en dos subgrupos.

Son más lentos de calcular que los jerárquicos aglomerativos

A continuación se muestra un ejemplo del método

jerárquico divisivo usando la función **Diana** de la librería **cluster**.

```
> bupadiv<-diana(bupa[,1:6],metric='euclidean')
```

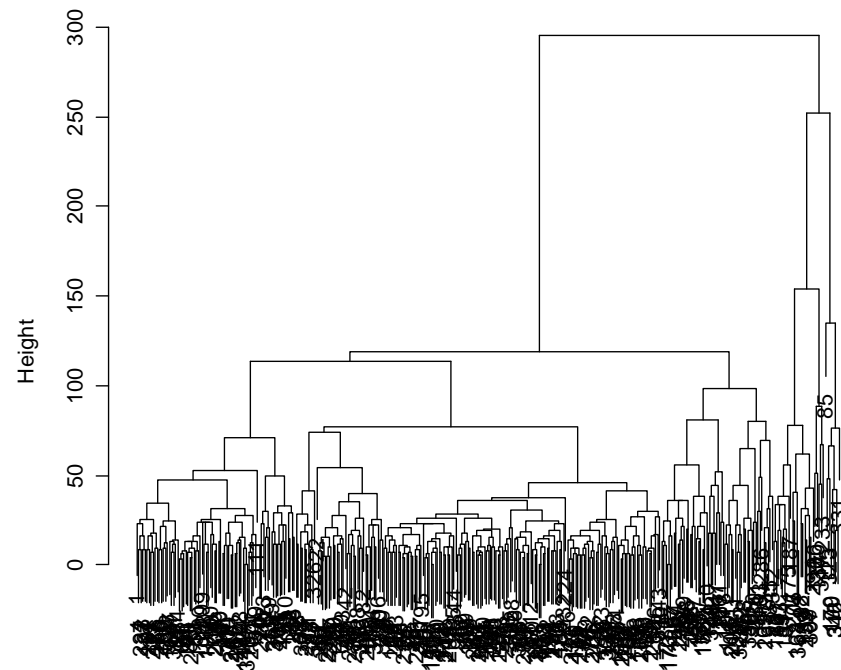
```
> plot(bupadiv,which=2)
```

```
> bupadiv
```

Ejemplo usando Diana

```
bupadiv=diana(bupa[,1:6],metric='euclidean')
plot(bupadiv,which=2)
bupadiv
b=cutree(bupadiv,k=2)
table(b)
 1  2
33 312
table( b,bupa [,7])
> table(cutree(bupadiv,k=3))
 1  2  3
25  8 312
```

Dendrogram of `diana(x = bupa[, 1:6], metric = "euclidean")`



Divisive Coefficient = 0.96

Comparación de métodos de particionamiento con los métodos jerárquicos.

Los **métodos de particionamiento** tienen la ventaja de que **satisfacen un criterio de optimilidad** aunque sea aproximadamente.

Desventajas: **Necesitan un valor inicial** del número de clusters y toma **mucho tiempo** obtener los clusters.

Por otro lados **los métodos jerárquicos** tienen la ventaja que son **rápidos** de calcular sobre todo el aglomerativo.

Desventaja: La **rigidez que le da la estructura de árbol** (el llamado factor de anidamiento). Es difícil corregir lo que se hizo antes.

Validacion clusters

Indices Internos. Estadísticas basados en las sumas de cuadrados entre clusters y dentro de clusters. El número de clusters K es aquel que maximiza o minimiza uno de estos índices. (Milligan, G.W. & Cooper, M.C.). Entre los principales están el índice de Dunn, el Índice de Davies-Bouldin.

Ancho de silueta promedio.

Determinar el número de componentes de una mezcla de distribuciones es lo mismo que determinar el número de clusters. Así, que se pueden usar los criterios de **AIC** (Criterio de Información de Akaike) y **BIC** (Criterio de Información Bayesiano).

-
- Índice de Dunn (1974). La idea es identificar los clusteres que estan bien compactos y bien separados de los demas. Dada una particion de clusters donde c_i representa el i -esimo cluster de la particion, se define el indice de Dunn por

$$D = \min_{1 \leq i \leq n} \left\{ \min_{1 \leq j \neq i \leq n} \left\{ \frac{d(c_i, c_j)}{\max_{1 \leq k \leq n} (d'(c_k))} \right\} \right\}$$

donde $d(c_i, c_j)$ – es la distancia entre los clusters c_i , y c_j y $d'(c_k)$ representa la distancia intracluster del cluster c_k .

En numero optimo de clusters es aquel que maximiza D .
La libreria fpc tiene una funcion cluster.stats que calcula el indice de Dunn.

```
> disbupa=dist(bupa[,1:6])
> cutree(bupadiv,k=2)
> cluster.stats(disbupa,a,bupa[,7])$dunn
[1] 0.06806368
```

Silhouette plots

Los plots siluetas, (Rousseeuw 1987) pueden ser usados para:

- Seleccionar el número de clusters.
- Evaluar cuan bien han sido asignados las observaciones en los clusters.

El **ancho de la silueta** (**silhouette width**) de la i -ésima observación es definida por: $sil_i = (b_i - a_i) / \max(a_i, b_i)$

Donde, a_i denota la distancia promedio entre la observación i y todas las otras que están en el mismo cluster de i , y b_i denota la distancia promedio mínima de i a las observaciones que están en otros clusters.

El valor de la silhouette varia entre -1 y +1.

Características de los Silhouette plots

Las observaciones con **ancho de silueta grande** están bien **agrupadas** mientras aquellas con **ancho de silueta baja** tienden a estar ubicada en el medio de dos clusters.

Para un número de clusters dado K , **el ancho de silueta promedio** de la configuración de conglomerados será simplemente el promedio de sil_i sobre todas las observaciones. Es decir,

$$\bar{s} = \frac{\sum_i sil_i}{n}$$

Kaufman y Rousseeuw (1990) sugirieron estimar **el número óptimo de cluster K** para el cual el ancho de silueta promedio sea la mayor posible.

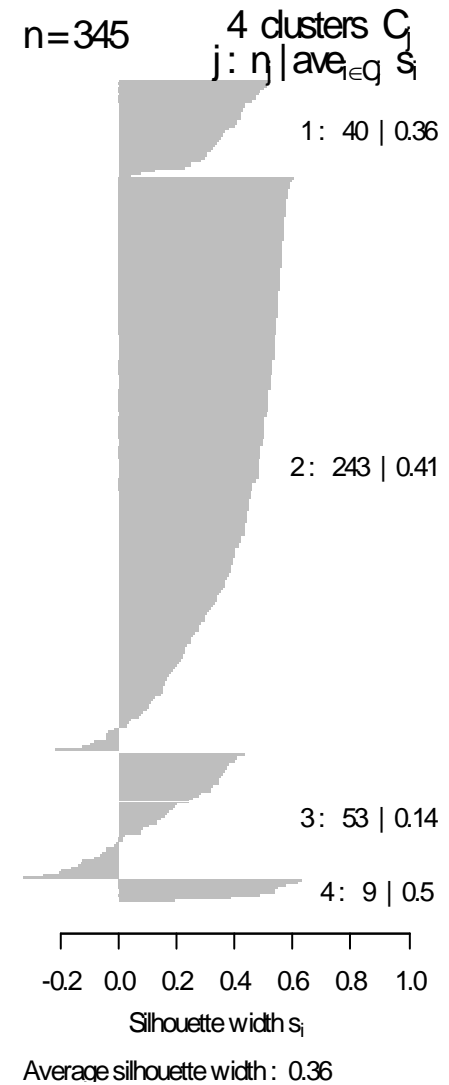
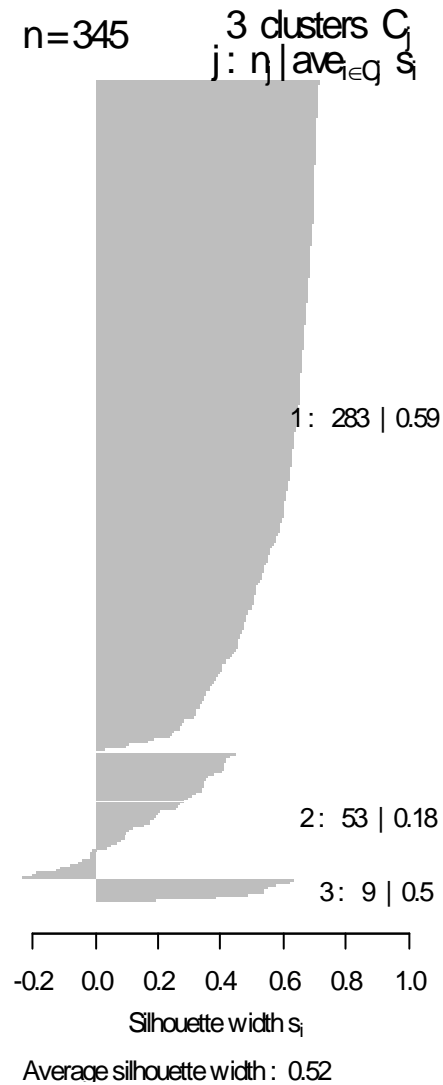
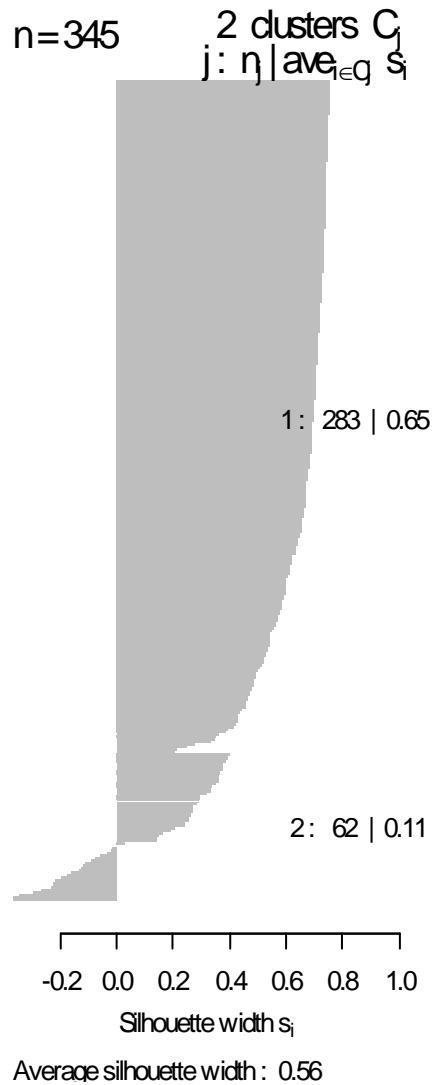
Ejemplo de los Silhouette plots

Ejemplo con el conjunto de datos **Bupa** y el metodo jerarquico aglomerativo

```
> agbupa=agnes(dist(bupa[,1:6]),method="ward")
> a=silhouette(cutree(agbupa,k=2),daisy(bupa[,1:6]))
> b=silhouette(cutree(agbupa,k=3),daisy(bupa[,1:6]))
> c=silhouette(cutree(agbupa,k=4),daisy(bupa[,1:6]))
> par(mfrow=c(1,3))
> plot(a,main="")
> plot(b, main="")
> plot(c,main="")
```

Mirando el plot k=2 clusters es lo recomendado.

PLOTS SILUETAS PARA CLUSTERING DE BUPA



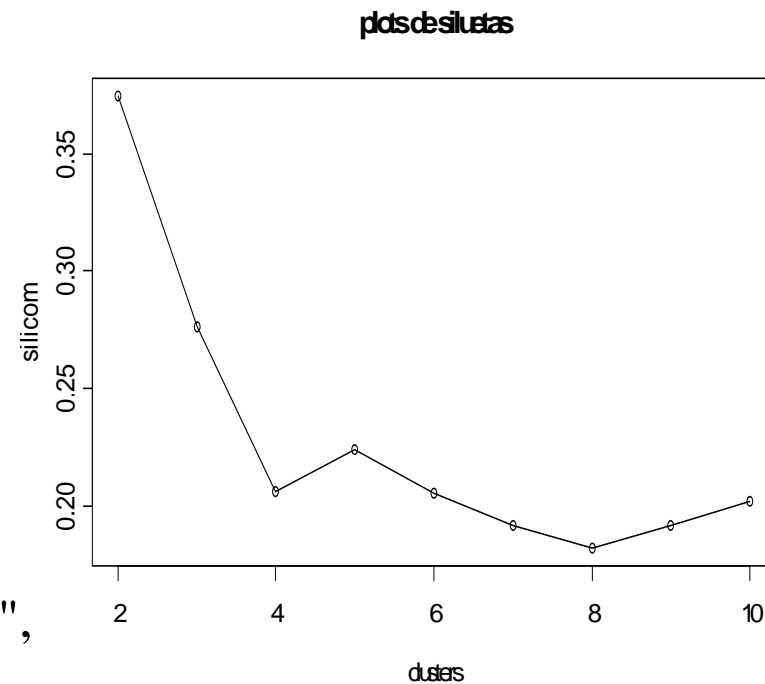
Ejemplo de los Silhouette plots

Ejemplo con el conjunto de datos Bupa y el metodo PAM

```
silicom=rep(0,9)
for(i in 1:9){
  silicom[i]=pam(vehicle[,1:18],i+1,
  diss=F,stand=T)$silinfo$avg.width}
```

```
plot(2:10,silicom)
plot(2:10,silicom,type="o",xla="clusters",
  main="plots de siluetas")
```

Aqui tambien salen 2 clusters.



Indices externos

Supongamos que tenemos dos particiones de n objetos $\mathbf{x}_1, \dots, \mathbf{x}_n$: la partición en R clases $U = \{u_1, \dots, u_R\}$ y la partición en C -clases $V = \{v_1, \dots, v_C\}$, por lo general una de ellas conocida de antemano.. Los índices externos de concordancia entre las particiones pueden ser expresados en término de una tabla de contingencia con entradas n_{ij} que representa el número de objetos que están en ambos clusters u_i and v_j , $i = 1, \dots, R$, $j = 1, \dots, C$. Sean

$$n_{i.} = \sum_{j=1}^C n_{ij} \quad \text{y} \quad n_{.j} = \sum_{i=1}^R n_{ij}$$

que denotan las sumas de filas y columnas de la tabla de contingencia. Sea

$$Z = \sum_{i=1}^R \sum_{j=1}^C n_{ij}^2$$

Dada las dos particiones U y V

a: numero de pares objetos que estan en el mismo cluster tanto en U como en V.

c: numero de pares de objetos que estan en el mismo cluster en V pero no en U.

b: numero de pares de objetos que estan en el mismo cluster en U pero no en V.

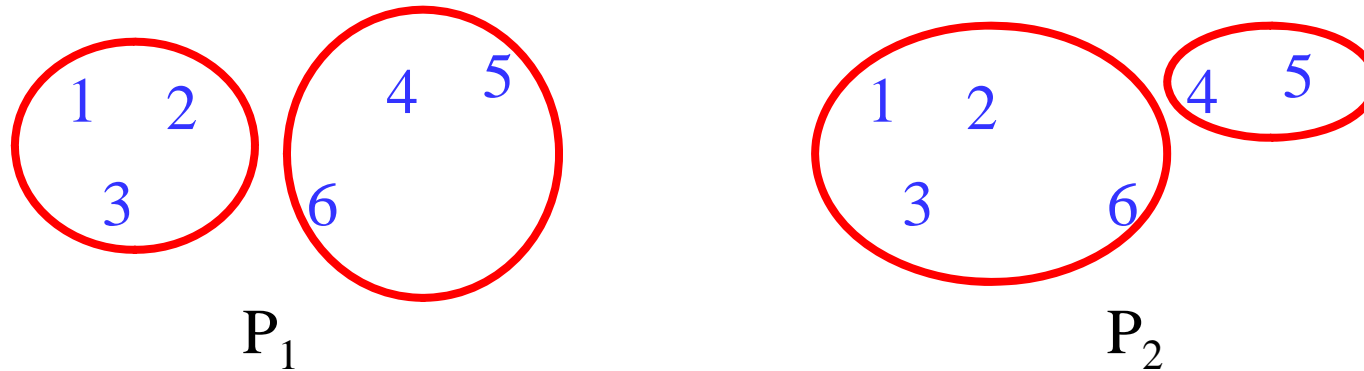
d: numero de pares de objetos que estan en diferentes clusters tanto en U como en V.

$m_1 = a + b = \sum_{i=1}^R \binom{n_i}{2}$ numero de pares de objetos en el mismo cluster en U.

$m_2 = a + c = \sum_{j=1}^C \binom{n_j}{2}$ numero de pares de objetos en el mismo cluster en V.

Ejemplo

$a=4, b=2, C=3, d=6, m_1=6, m_2=7, Z=14$



Notar que $M=a+b+c+d= \binom{n}{2}$

- Rand(1971)

$$Rand = 1 + \frac{(z - (1/2)(\sum_{i=1}^R n_i^2 + \sum_{j=1}^C n_j^2))}{\binom{n}{2}} = \frac{a+d}{\binom{n}{2}}$$

- Jaccard

$$Jac = \frac{(z-n)}{\sum_{i=1}^R n_i^2 + \sum_{j=1}^C n_j^2 - Z - n} = \frac{d}{b+c+d}$$

- Fowlkes and Mallows

$$FM = \frac{(1/2)(z-n)}{[\sum_{i=1}^R \binom{n_i}{2} \sum_{j=1}^C \binom{n_j}{2}]^{1/2}} = \frac{d}{\sqrt{m_1 m_2}}$$

Un valor de Rand, Jaccard y FM cercano a 1 indica un buen agrupamiento.

La medida Γ de Hubert y Arabie (1985)

Sean las variables aleatorias:

$X(i,j)=1$ si los objetos i y j caen en el mismo cluster de la particion U e igual a 0 en otro caso.

$Y(i,j)=1$ si los objetos i y j caen en el mismo cluster de la particion V e igual a 0 en otro caso.

Se define la medida Γ de Hubert como:

$$\Gamma = \frac{1}{M} \sum \sum X(i, j)Y(i, j)$$

Para obtener valores de Γ entre -1 y 1 se prefiere normalizarlo y se obtiene

$$\Gamma = [(1/M) \sum \sum (X(i, j) - \bar{X})(Y(i, j) - \bar{Y})] / S_X S_Y$$

Que es equivalente a:

$$H = [\sum \sum X(i, j)Y(i, j) / M - \bar{X}\bar{Y}] / S_X S_Y$$

Pero, usando el hecho que X y Y son binomiales, se tiene que

$$\bar{X} = \frac{\sum_{i=1}^R \binom{n_{i.}}{2}}{M} \quad \bar{Y} = \frac{\sum_{J=1}^C \binom{n_{.J}}{2}}{M}$$

Que son las probabilidades de que los objetos i y j caigan en los mismos clusters de las particiones U y V respectivamente

Similarmente,

$$S_X = \bar{X}(1 - \bar{X}) = \frac{\sum_{i=1}^R \binom{n_{i.}}{2}}{M} \left(1 - \frac{\sum_{i=1}^R \binom{n_{i.}}{2}}{M}\right)$$

$$S_Y = \bar{Y}(1 - \bar{Y}) = \frac{\sum_{j=1}^C \binom{n_{.j}}{2}}{M} \left(1 - \frac{\sum_{j=1}^C \binom{n_{.j}}{2}}{M}\right)$$

Finalmente,

$$\frac{\sum \sum X(i, j)Y(i, j)}{M} = \frac{\sum_{i=1}^R \sum_{j=1}^C \binom{n_{ij}}{2}}{M} = \frac{\sum_{i=1}^R \sum_{j=1}^C n_{ij}^2 - \sum_{i=1}^R \sum_{j=1}^C n_{ij}}{2M} = \frac{Z - n}{2M}$$

Sustituyendo, en la forma de Γ normalizada se tiene

$$H = \frac{.5 * M (z - n) - \sum_{i=1}^R \binom{n_{i.}}{2} \sum_{j=1}^C \binom{n_{.j}}{2}}{\sqrt{\sum_{i=1}^R \binom{n_{i.}}{2} \sum_{j=1}^C \binom{n_{.j}}{2} (M - \sum_{i=1}^R \binom{n_{i.}}{2}) (M - \sum_{j=1}^C \binom{n_{.j}}{2})}}$$

Ahora usando las identidades

$$a + b = \sum_{i=1}^R \binom{n_{i.}}{2} \quad a + c = \sum_{j=1}^C \binom{n_{.j}}{2}$$

y $a = (z-n)/2$, se tiene la siguiente formula simplificada de H

$$H = \frac{Ma - (a+b)(a+c)}{\sqrt{(a+b)(a+c)(M - (a+b))(M - (a+c))}}$$

```
> agbupa = agnes(dist(bupa[, 1:6]), method = "ward")
> a = cutree(agbupa, k = 2)
> mexter(bupa[, 7], a)
$rand
[1] 0.4990563

$jaccard
[1] 0.4163361

$fandm
[1] 0.5954606

$hubert
[1] -0.01218121
```

```
> agbupa=agnes(dist(bupa[,1:6]),method="complete")
> c=cutree(agbupa,k=2)
> mexter(bupa[,7],c)
$rand
[1] 0.5103809
$jaccard
[1] 0.5091815
$fandm
[1] 0.7124241
$hubert
[1] -0.01026507
> table(c)
c
 1  2
344 1
```

-
- FOM=Figure of Merit (Yeung and Ruzzo,2001)
 - The Gap Statistics (Tibshirani,2000).
 - **Clest (Dudoit & Fridlyand, 2002)**

Algoritmo Clest (Dudoit & Fridlyand, 2002).

Estima el número de conglomerados basado en la precisión de la predicción.

Para cada número de clusters k , se divide al azar B veces el conjunto de datos original en dos conjuntos que no se superponen.

Uno de ellos L_b forma la muestra de entrenamiento y el otro, T_b forma la muestra de prueba, $b = 1, \dots, B$.

-Aplicar el algoritmo de conglomerados (se recomienda el PAM) a las observaciones en el conjunto de entrenamiento L_b .

- Construir un clasificador (puede ser LDA, k-nn, CART, etc) usando las

etiquetas obtenidas del método de conglomerados.

-Aplicar el clasificador al conjunto de prueba T_b .

-Aplicar el algoritmo de conglomerados al conjunto de prueba T_b .

-
- Calcular un score $s_{k,b}$ comparando las etiquetas del conjunto de prueba obtenidas por conglomerados y por predicción del clasificador. Estos scores se obtienen aplicando índices externos como RAND, Jaccard o Fowkles y Mallows (FM)
 - El *score de similaridad* para los k clusters es la mediana de los B scores de similaridad $t_k = \text{median}(s_{k,1}, \dots, s_{k,B})$.
 - El *número de clusters* K es estimado comparando el score observado t_k con su valor esperado asumiendo cierta distribución de Referencia.

El programa Machaon (2004)

- Nadia.Bolshakova (CS Department, Trinity College, UK)

Cluster Validity Tool for gene Expression data

Clustering basados en modelos (Fraley y Raftery, JASA 2002)

Analisis de conglomerados (cluster) es el agrupamiento de objetos en grupos bien coherentes basado en características medidas en los objetos. Fue introducido a los finales de 1950 por Sokal, Sneath y otros, y se ha desarrollado principalmente usando metodos heuristicos. Recientemente se ha encontrado que analisis de clusters basado en modelos probabilisticos tanto para entender los metodos actuales de hacer clustering como par sugerir nuevos metodos. El uso de modelo tambien permite responder interrogantes tales como, cuantos clusters usar?, que metodo de cluster es el mas conveniente? Como tratar la presencia de outliers?

Clustering basado en modelos

Desde hace tiempo, investigadores, se han dado cuenta que análisis de conglomerados puede ser llevado a cabo usando modelos de probabilidad. Con estos modelos se está tratando de ver cuando es que un cierto método de clustering funciona bien.

Se ha demostrado que algunos de los métodos heurísticos de hacer conglomerados son simplemente métodos de estimación aproximados de modelos de probabilidad. Por ejemplo, el método de k-means y el método de Ward son equivalentes a conocidos métodos para maximizar aproximadamente la clasificación usando una normal multivariada cuando la matriz de covarianza es la misma para cada componente y proporcional a la matriz identidad.

Clustering basado en mezclas finitas

Modelos de mezcla finitas han sido propuestos y estudiado a menudo en el contexto de clasificacion (Wolfe, 1963, 185, 1967, 1970; Edwards y Cavalli-Sforza 1965; Day 1969; Scott y Symons 1971; Duda y Hart 1973; Binder 1978).

In modelos de mezclas finita cada componente de la distribucion de probabilidad corresponde a un cluster.

El problema de determinar el numero de componentes puede ser reformulado como un problema de seleccion de modelos

Los outliers son tratados mediante la adicion de una o mas componente representando una distribucion distinta para los datos anomalos.

Clustering basado en Mezclas finitas

La funcion de likelihood de un modelo de mezcla con G componentes dado que se observo el la muestra aleatoria y_1, y_2, \dots, y_n de la variable aleatoria y esta definida por

$$L(\theta_1, \dots, \theta_G; \tau_1, \dots, \tau_G / y) = \prod_{i=1}^n \sum_{k=1}^G \tau_k f_k(y_i / \theta_k)$$

Donde f_k y θ_k son las funciones de densidad y los parametros de la k-esima componente de la muestra y τ_k es la probabilidad de que una observacion pertenezca a la k-esima componente . Los τ_k son no negativos y su suma debe dar 1.

Por lo general f_k es una densidad normal multivariada ϕ_k parametrizada por su media y matriz de covarianza.

La distribución Normal Multivariada

$$\phi_k(y_i / \mu_k, \Sigma_k) \equiv \frac{\exp\left(-\frac{1}{2}(y_i - \mu_k)^T \Sigma_k^{-1}(y_i - \mu_k)\right)}{\sqrt{\det(2\pi\Sigma_k)}}$$

Clustering basado en mezclas Gaussianas

Las características geométricas (forma, volumen, orientación) de los clusters son determinados por las covarianzas Σ_k , que a su vez pueden ser parametrizadas para imponer restricciones entre clusters. Así, si se considera $\Sigma_k = \lambda I$, entonces todos los clusters son esféricos y del mismo tamaño, si $\Sigma_k = \Sigma$ entonces todos los clusters tienen la misma geometría pero no son necesariamente esféricos. En el primer caso se necesita solo un parámetro y en el segundo $d(d+1)/2$ parámetros.

El criterio BIC (Bayesian Information Criterio) para seleccionar el mejor modelo

$$2 \log p(D / M_k) \approx 2 \log p(D / \hat{\theta}_k, M_k) - v_k \log(n) = BIC_k$$

donde v_k es el numero de parametros independientes a ser estimado en el modelo M_k (Schwarz 1978).

El mejor modelo sera aquel tiene el BIC ma grande

Uso de Mclust

Los siguientes modelos son comparados en 'Mclust':

"EII": spherical, equal volume

"VII": spherical, unequal volume

"EEI": diagonal, equal volume, equal shape

"VEI": diagonal, varying volume, equal shape

"EVI": diagonal, equal volume, varying shape

"VVI": diagonal, varying volume, varying shape

"EEE": ellipsoidal, equal volume, shape, and orientation

EEV": ellipsoidal, equal volume and equal shape

"VEV": ellipsoidal, equal shape

"VVV": ellipsoidal, varying volume, shape, and orientation

El comportamiento de los modelos dependen de la descomposicion espectral de las matrices de covarianzas

```
> a=Mclust(bupa[,1:6],1:10)
> a
best model: diagonal, varying volume and shape with 4
  components
> a$bic
[1] -14867.11
> table(a$class)

 1  2  3  4
135 152 53  5
> a$parameters$pro
[1] 0.40769889 0.41884788 0.15928831 0.01416492
```

```
> a$parameters$mean
```

```
      [,1] [,2] [,3] [,4]  
V1 90.429894 89.313426 91.110518 96.69474  
V2 73.818404 64.056493 73.900558 82.77228  
V3 29.045119 20.299498 57.676037 61.74509  
V4 24.067698 19.222119 38.203606 49.03473  
V5 35.667890 16.089920 88.777391 202.03921  
V6  3.700042  2.163306  5.584584 10.65408
```

> a\$BIC

	EII	VII	E EI	VEI	EVI	VVI	EEE	EEV	VEV	VVV
1	-18303.43	-18303.43	-15983.15	-15983.15	-15983.15	-15983.15	-15568.76	-15568.76	-15568.76	-15568.76
2	-17202.80	-16598.74	-15510.36	-15132.42	-15321.70	-15001.33	-15441.96	-15297.84	-14983.84	-14964.50
3	-16838.22	-16197.67	-15409.54	-14992.76	-15209.54	-15040.37	-15318.12	-15277.09	-15036.85	NA
4	-16678.34	-16008.67	-15408.63	-14940.84	-15185.15	-14867.11	-15314.72	-15309.64	-15097.61	NA
5	-16554.47	-15886.13	-15445.39	-14957.04	-15161.28	-14894.18	-15245.00	-15330.92	-15155.63	NA
6	-16468.54	-15783.68	-15158.72	-14974.77	-15181.85	NA	-15154.37	-15316.74	-15193.14	NA
7	-16466.85	-15776.61	-15194.85	-14976.11	NA	NA	-15172.70	-15406.73	-15178.47	NA
8	-16470.31	-15760.68	-15183.71	-15009.91	NA	NA	-15207.83	-15539.95	-15316.13	NA
9	-16097.65	-15749.74	-15222.67	-15033.18	NA	NA	-15208.77	-15566.15	-15350.26	NA
10	-16090.24	-15726.55	-15199.41	-15039.04	NA	NA	-15230.21	-15620.28	-15527.11	NA

plot(a,bupa[,1:6])

