Análisis cluster

Guillermo Ayala Gallego

Análisis cluster

Guillermo Ayala Gallego

2024-04-23

Un ejemplo artificial: datos Ruspini

```
library(cluster)
data(ruspini)
```

...

Un ejemplo con muestras

```
library(multtest)
data(golub)
grep("CCND3 Cyclin D3",golub.gnames[,2])
```

```
[1] 1042
```

```
grep("Zyxin",golub.gnames[,2])
```

```
[1] 2124
```

```
cz.data = data.frame(golub[1042,],golub[2124,])
colnames(cz.data) = c("CCND3 Cyclin D3","Zyxin")
```

Disimilaridades

Distancias

- Distancia euclídea

\[d(x,y) = \sqrt{\sum_{k=1}^d (x_k -y_k)^2} \]

- Distancia de Manhattan \[d(x,y) = \sum_{k=1}^d |x_k - y_k| \]

Correlación de Pearson

- Supongamos que cuando evaluamos si dos genes están **próximos** interpretamos que están corregulados.
- Una medida de **co**rrelación es \[r = \frac{\sum_{i=1}^n (x_i - \bar{x}_n) (y_i - \bar{y}_n)} {\sqrt{\sum_{i=1}^n (x_i - \bar{x}_n)^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y}_n)^2}}. \]

...

...

```
cor(x1,y1)
```

```
[1] 0.8490916
```

Una correlación mayor

Y tenemos una correlación observada de

```
cor(x1,y1)
```

```
[1] 0.992855
```

Correlación muy pequeña

...

Y la correlación muestral es

```
cor(x1,y1)
```

```
[1] 0.08857924
```

Disimilaridades utilizando coeficientes de correlación

- \(d(x,y) = 1 - r_{x,y}\),
- \(d(x,y) = (1 - r_{x,y})/2\),
- \(d(x,y) = 1 - |r_{x,y}|\),
- \(d(x,y) = \sqrt{1 - r_{x,y}^2}\).

Evaluando las disimilaridades

- Si la disimilaridad que estamos considerando es simétrica entonces \[d(x_i,x_j) = d(x_j,x_i) \]
- D es una matriz simétrica

- Además \(d(x_i, x_i) = 0\).
- Se almacena una parte de la matriz (superior o inferior).

Calculando disimilaridades

- Distancia euclídea

```
cz.dist.euclidea = dist(cz.data,method="euclidian")
class(cz.dist.euclidea)
```

```
[1] "dist"
```

- Distancia de Manhattan

```
cz.dist.manhattan = dist(cz.data,method="manhattan")
```

- Basadas en correlación

```
data(golub,package="multtest")
golub.cor1 = 1-cor(t(golub))
golub.cor2 = (1-cor(t(golub)))/2
golub.cor3 = 1-abs(cor(t(golub)))
golub.cor4 = sqrt(1-cor(t(golub))^2)
```

Disimilaridades entre grupos de observaciones

- Supongamos que tenemos un banco de datos con \(n\) individuos cuyos índices son \(\{1, \ldots, n\}\).
- Sean \(A\) y \(B\) dos subconjuntos disjuntos del conjunto de índices de la muestra \(\{1, \ldots, n\}\)
- Si denotamos la disimilaridad entre \(A\) y \(B\) como \(d(A,B)\) entonces
  - **Enlace simple**: \(d(A,B) = \min_{a \in A, b \in B} d(a,b)\).
  - **Enlace completo**: \(d(A,B) = \max_{a \in A, b \in B} d(a,b)\).
  - **Promedio**: \(d(A,B) = \frac{1}{|A| \times |B|} \sum_{a \in A, b \in B} d(a,b)\) siendo \(|A|\) es el cardinal del conjunto \(A\).

¿Qué es una clasificación?

- Sea \(\{1, \ldots, n\}\) el conjunto de índices que indexan las distintas observaciones.
- Supongamos que \(\{C_1, \ldots, C_r\}\) es una partición de este conjunto de índices:
  1. \(C_i \subset \{1, \ldots, n\}\); son disjuntos dos a dos, \(C_i \cap C_j= \emptyset\) si \(i \neq j\) con \(i,j = 1, \ldots, n\) y
  2. \(\cup_{i=}^r C_i = \{1, \ldots, n\}\).

Cluster jerárquico aglomerativo

1. Tenemos grupos unitarios formados por cada una de las observaciones. Tenemos pues una partición inicial \(C_i=\{i\}\) con \(i=1,\ldots,n\). En un principio, cada dato es un grupo.
2. Calculamos las disimilaridades entre los elementos de la partición. Para ello utilizamos cualquiera de los procedimientos antes indicados.
3. Agrupamos los dos conjuntos de la partición más próximos y dejamos los demás conjuntos igual. Tenemos ahora \(C_i\) con \(i=1,\ldots,k\).
4. Si tenemos un solo conjunto en la partición paramos el procedimiento.
5. Volvemos al paso **2**.

agnes: dendograma

Dendograma para un cluster jerárquico aplicado a los datos ruspini con métrica euclídea y el promedio para la disimilaridad entre grupos.

```
ruspini.ag = agnes(ruspini,metric = "euclidean",method="average")
```

```
ruspini.ag = agnes(ruspini,metric = "euclidean",method="average")
```

...

Clasificación

- Supongamos que decidimos quedarnos con cuatro grupos.
- Las clasificaciones de los datos son las siguientes donde la etiqueta que se asigna a cada grupo es arbitraria.

```
cutree(ruspini.ag,4)
```

```
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[39] 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
```

...

Clasificación obtenida para los datos ruspini utilizando un cluster jerárquico con distancia euclídea, promedio entre grupos y cuatro grupos.

```
plot(ruspini,col=cutree(ruspini.ag,4),pch=cutree(ruspini.ag,4))
```

Métodos de particionamiento

Características

- Suponemos ahora que tenemos una idea de cuántos grupos hay.
- Posiblemente hemos realizado un análisis jerárquico previo con todos los datos o, si eran muchos, con una selección aleatoria de los datos.
- En principio, vamos a suponer que fijamos el número de grupos a considerar.

- Suponemos pues que **sabemos** el número de grupos y lo denotamos por \(k\).

Método de las k-medias

- Supongamos que tenemos \(C_1, \ldots, C_k\) una partición de \(\{1, \ldots, n\}\).
- Un modo bastante natural de valorar la calidad del agrupamiento que la partición nos indica sería la siguiente función: \[\sum_{i=1}^k \sum_{j \in C_i} d_E(x_j,\bar{x}_{C_i})^2, \] donde \(d_E\) denota aquí la distancia euclídea y \[\bar{x}_{C_i} = \frac{1}{|C_i|} \sum_{j \in C_i} x_j, \] es el vector de medias del grupo cuyos índices están en \(C_i\).
- Una partición será tanto mejor cuanto menor sea el valor de la función anterior.
- El procedimiento de agrupamiento de las k-medias simplemente se basa en elegir como partición de los datos aquella que nos da el **mínimo** de la función objetivo.

kmeans

```
ruspini.km = kmeans(ruspini,4)
```

La clasificación viene dada por

```
ruspini.km$cluster
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
 4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  4  3  3  3  3  3  3
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
 3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3  3
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
 3  3  3  3  3  3  3  3  1  1  2  1  1  1  1  1  1  2  2  2  2  2  2
```

Particionamiento alrededor de los mediodes (PAM)

- Supongamos que tomamos \(k\) individuos de la muestra que denotamos por \(m_i\) con \(i=1,\ldots,k\).
- Particionamos la muestra en \(k\) grupos de modo que el grupo \(C_i\) está formado por los individuos más próximos a \(m_i\) que a cualquier otro \(m_j\) con \(j \ne i\), \[C_i = \{l: d(l,i) = \min_{j \ne i} d(l,j) \}. \]
- Consideremos la siguiente cantidad: \[\sum_{i=1}^k \sum_{j \in C_i} d(j,m_i). \]
- En el método de particionamiento alrededor de los mediodes nos planteamos encontrar las observaciones \(m_1,\ldots,m_k\) que minimizan el valor anterior.

pam

```
ruspini.pam = pam(ruspini,4)
```

...

```
plot(ruspini.pam,which=1)
```

Silueta

- Para la observación \(i\) y el grupo \(C\) consideramos \[\bar{d}(i,C) = \frac{1}{|C|} \sum_{j \in C} d(i,j). \]
- Para cada observación \(i\), sea \(A\) su cluster y \(a(i) = \bar{d}(i,A)\).
- Consideremos \[b(i) = \min_{C \ne A} \bar{d}(i,C). \]

- Definimos \[s(i) = 1 - \frac{a(i)}{b(i)} \text{ si} a(i) < b(i),\] \[s(i) = 0 \text{ si } a(i) = b(i),\] \[s(i) = \frac{b(i)}{a(i)} - 1 \text{ si } a(i) > b(i).\]
- O equivalentemente \[s(i) = \frac{b(i) - a(i)}{\max\{a(i),b(i)\}}\]

Anchura media de la silueta

¿Cómo interpretar \(\bar{s} = \sum_{i=1}^n s_i /n\)?

\(0.71-1.00\)	Fuerte estructura
\(0.51-0.70\)	Estructura razonable
\(0.26-0.50\)	Estructura débil. Probar otros métodos
\(\leq 0.25\)	No se encuentra estructura

...

```
ruspini.pam = pam(ruspini,4)
summary(silhouette(ruspini.pam))
```

```
Silhouette of 75 units in 4 clusters from pam(x = ruspini, k = 4) :
 Cluster sizes and average silhouette widths:
       20        23        17        15
0.7262347 0.7548344 0.6691154 0.8042285
Individual silhouette widths:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.4196  0.7145  0.7642  0.7377  0.7984  0.8549
```

...

```
plot(silhouette(ruspini.pam))
```

Análisis cluster y datos de expresión de gen

- Hemos visto algunos métodos de clasificación así como de valoración de hay estructura cluster en un conjunto de datos dados.
- Estos métodos se aplican (sin modificación) en contextos muy diversos.
- ¿Qué interés particular tiene su aplicación para datos de expresión de gen?
- Lo primero a considerar es que nos podemos plantear o bien la clasificación de las filas de la matriz de expresión o bien la clasificación de las columnas.
- Si clasificamos las filas lo que tenemos en cada fila es el perfil de expresión del gen en todas las muestras del estudio.
- Si clasificamos las columnas, las muestras, entonces podemos estar, por ejemplo, buscando subclases de células tumorales.
- Los resultados de la clasificación de genes y muestras puede utilizarse como control de calidad. Sobre todo en las muestras tenemos un diseño experimental previo (habitualmente, el diseño grupo control frente a grupo de tratamiento).

GSE20986

- Cargamos los datos.

```
data(gse20986,package="tamidata")
```

- Seleccionamos genes.

```
library(genefilter)
gse0 = nsFilter(gse20986,var.func = mean, var.cutoff = 0.7)
gse1 = nsFilter(gse0$eset,var.func = IQR, var.cutoff = 0.7)
gse = gse1$eset
```

- ¿Con cuántas filas nos hemos quedado?

```
dim(exprs(gse))
```

```
[1] 1874   12
```

...

- Vamos a clasificar las muestras y compararemos con la clasificación original de las mismas.
- Utilizamos distancia Manhattan y promedio con el método PAM.
- Primero calculamos la matriz de distancias.

```
d0 = dist(t(exprs(gse)),method = "manhattan")
```

- Y aplicamos PAM.

```
library(cluster)
gse.pam = pam(d0,diss = TRUE,k=4)
```

- La clasificación original es

```
tipo0 = rep(1:4,rep(3,4))
```

- Comparamos la clasificación original con la obtenida.

```
table(tipo0,gse.pam$clustering)
```

```
tipo0 1 2 3 4
    1 1 1 1 0
    2 2 1 0 0
    3 3 0 0 0
    4 0 0 0 3
```

- ¿Qué significa esto?

Heatmap

```
heatmap(exprs(gse),hclustfun=agnes)
```