

# Modelos Estadísticos para la Regresión y la Clasificación

## Clase 11: Clustering

Mathias Bourel

Instituto de Matemática y Estadística Prof. Rafael Laguardia (IMERL)  
Facultad de Ingeniería, Universidad de la República, Uruguay

30 de septiembre de 2024

# Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
  - Algoritmo de K-medias.
  - Una variante robusta: PAM
- 4 Técnicas jerárquicas.
  - Métodos aglomerativos.
  - Métodos divisivos.
- 5 Spectral Clustering
- 6 Comparación de particiones

El objetivo del análisis de clusters es el agrupar elementos (datos, individuos, variables, etc.) en grupos (clusters) homogéneos, en función de las similitudes o diferencias entre ellos. Se arma grupos de observaciones que sean lo más parecidas entre ellas pero también lo más diferentes que observaciones de los otros grupos.

El objetivo del análisis de clusters es el agrupar elementos (datos, individuos, variables, etc.) en grupos (clusters) homogéneos, en función de las similitudes o diferencias entre ellos. Se arma grupos de observaciones que sean lo más parecidas entre ellas pero también lo más diferentes que observaciones de los otros grupos.

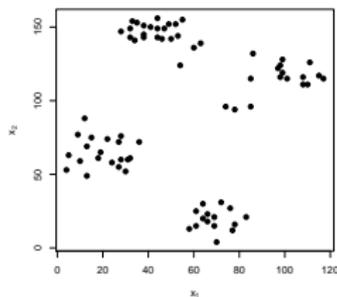
El análisis de cluster es un método de *aprendizaje no supervisado* donde se dispone únicamente de valores de  $X$  y no hay etiquetas de clase que identifiquen las observaciones.

El objetivo del análisis de clusters es el agrupar elementos (datos, individuos, variables, etc.) en grupos (clusters) homogéneos, en función de las similitudes o diferencias entre ellos. Se arma grupos de observaciones que sean lo más parecidas entre ellas pero también lo más diferentes que observaciones de los otros grupos.

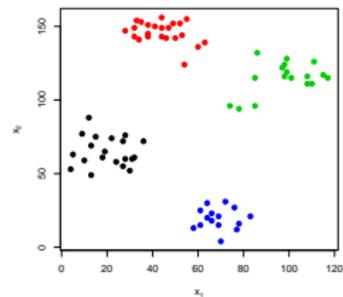
El análisis de cluster es un método de *aprendizaje no supervisado* donde se dispone únicamente de valores de  $X$  y no hay etiquetas de clase que identifiquen las observaciones.

A diferencia de los problemas de clasificación, la (posible) estructura de los grupos es desconocida a priori, incluyendo el número de clases o clusters.

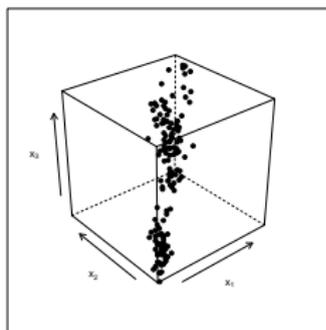
- Posibles aplicaciones son:
  - en marketing, para segmentar el mercado en pequeños grupos homogéneos donde realizar campañas publicitarias específicas;
  - en biología, para dividir organismos en estructuras jerárquicas con el propósito de describir la diversidad biológica;
  - en medicina, para diseñar tratamientos específicos para distintos grupos de riesgo;
  - en psicología, para clasificar individuos en distintos tipos de personalidad, etc.
- Las técnicas de clustering esencialmente intentan extender y formalizar lo que los seres humanos observan muy bien en dos o tres dimensiones.



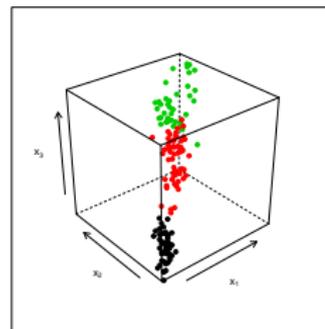
**Figura:** Ejemplo en dos dimensiones donde los grupos parecen “naturales”.



**Figura:** Grupos obtenidos mediante un algoritmo de clustering (PAM).



**Figura:** Ejemplo en tres dimensiones sobre los datos iris.



**Figura:** Grupos obtenidos mediante un algoritmo de clustering (PAM).

- Describiremos técnicas “más automáticas” de clustering dado que son las más apropiadas para trabajar problemas en *grandes dimensiones* (número elevado de observaciones y/o variables).
- Nos concentraremos especialmente en estas técnicas:
  - de *particionamiento* (K-medias, PAM),
  - *jerárquicas* (aglomerativas, divisivas, Ward),
  - *basadas en modelos*,
  - *Spectral Clustering*.
- Un aspecto central en todas ellas es la noción de *similaridad* o *disimilaridad* entre los objetos a ser agrupados. Para ello es necesario en muchos casos utilizar medidas de distancia entre los datos.

# Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
  - Algoritmo de K-medias.
  - Una variante robusta: PAM
- 4 Técnicas jerárquicas.
  - Métodos aglomerativos.
  - Métodos divisivos.
- 5 Spectral Clustering
- 6 Comparación de particiones

- Una distancia  $d : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$  cumple:
  - 1  $d(x, y) \geq 0, \forall x, y \in \mathbb{R}^p$  ;  $d(x, y) = 0 \Leftrightarrow x = y$
  - 2  $d(y, x) = d(x, y) \forall x, y \in \mathbb{R}^p$
  - 3  $d(x, y) \leq d(x, z) + d(z, y) \forall x, y, z \in \mathbb{R}^p$ .
- Para variables cuantitativas, las distancias entre  $x_i = (x_{i1}, \dots, x_{ip})$  y  $x_j = (x_{j1}, \dots, x_{jp}) \in \mathbb{R}^p$  más utilizadas son:
  - *Euclídea* o  $L_2$ :  $d(x_i, x_j) = (\sum_{k=1}^p (x_{ik} - x_{jk})^2)^{1/2}$ ,
  - *Manhattan* o  $L_1$ :  $d(x_i, x_j) = \sum_{k=1}^p |x_{ik} - x_{jk}|$ ,
  - *Minkowski* o  $L_q$  ( $q \geq 1$ ):  $d(x_i, x_j) = (\sum_{k=1}^p (x_{ik} - x_{jk})^q)^{1/q}$
- La distancia euclídea es la más utilizada en la práctica. La distancia  $L_1$  es más robusta frente a la presencia de datos atípicos (*outliers*). La distancia de Minkowski es una generalización de las otras dos.
- Estas medidas de distancia no son invariantes frente a cambios de escala de las variables. Por eso, buscaremos estandarizar los datos previamente (media cero y variancia 1).
- Una disimilaridad es una función  $d : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$  que verifica las propiedades 1 y 2 de distancia.

```
> (x1 <- c(-1, 2, 3.5, sqrt(2), -5))
[1] -1.000000 2.000000 3.500000 1.414214 -5.000000
> (x2 <- c(2, 8, 6, pi, -5))
[1] 2.000000 8.000000 6.000000 3.141593 -5.000000
>
> # distancia 'euclidea' o L2
> sqrt(sum( (x1 - x2)^2 ))
[1] 7.364363

> # distancia 'Manhattan' o L1
> sum( abs(x1 - x2) )
[1] 13.22738

> # distancia 'Minkowski' o Lq
> q <- 2.5 # por ejemplo
> (sum( (abs(x1 - x2))^q ))^(1/q)
[1] 6.731693
```

- Existen otras distancias, como la de *Mahalanobis*, para variables cuantitativas:

$$d(x, y) = (x - \mu)' \Sigma^{-1} (y - \mu)$$

donde  $\Sigma$  es la matriz de varianzas-covarianzas. La misma toma en cuenta las correlaciones entre las variables. Distintas distancias a menudo dan resultados diferentes (incluso para una misma técnica).

- Para otro tipo de variables (categóricas, ordinales o nominales), se utilizan por lo general otro tipo de distancias (basadas en el número de coincidencias o discordancias, por ejemplo) (Varmuza y Filmoser, 2009).
- La elección de la medida de distancia a utilizar puede ser tan o más importante que la técnica de cluster utilizada (Hastie y otros, 2009).
- Sin embargo, esta elección no es sencilla y requiere por lo general conocimiento específico del problema a resolver.
- Dadas  $n$  observaciones  $x_1, \dots, x_n \in \mathbb{R}^p$ , algunas técnicas de clustering requieren el cálculo de las distancias entre todo par de observaciones  $(x_i, x_j)$ .
- Esta información puede representarse mediante una matriz de distancias, de dimensión  $n \times n$ ,  $D = ((d_{ij}))$ , simétrica, donde  $d_{ij} = d(x_i, x_j)$ .

# Ejemplo numérico en R

```
> X <- matrix(sample(-5:5, 200, rep = T), ncol = 20)

> dim(X)
[1] 10 20

> head(X, 5) # primeras 5 filas

      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14]
[1,]    4    2    4    5   -4   -1   -1   -2    3    2   -1   -4   -4   -1
[2,]   -1    2    2    5   -3    5   -2   -4    3   -5   -1    4    0   -3
[3,]    4   -5    1   -3   -5    2    2    5    3    4   -3    2    5   -5
[4,]   -2   -1    5    4   -5   -4    0    2   -2   -5    3   -4   -3    0
[5,]   -3    5   -3   -2   -4   -2    2   -1    5    0    2    0   -4    3

      [,15] [,16] [,17] [,18] [,19] [,20]
[1,]   -2   -4   -1   -1    2   -4
[2,]   -5    0   -3    4   -1    3
[3,]    2   -3    3    4    2   -3
[4,]    0    4   -4   -2   -2    3
[5,]   -2   -3    3   -4   -3   -5

> round(dist(X, 'euclidean', diag = T), 3)

      1      2      3      4      5      6      7      8      9     10
1  0.000
2 17.776  0.000
3 19.748 22.494  0.000
4 17.578 18.628 24.960  0.000
5 16.371 21.307 22.847 20.664  0.000
6 19.000 20.952 19.570 22.450 21.331  0.000
7 19.442 19.799 20.396 20.712 20.248 21.424  0.000
8 14.491 19.545 25.219 21.190 16.613 20.372 23.409  0.000
9 19.287 21.260 21.166 20.421 19.647 27.368 20.543 21.260  0.000
10 18.138 22.694 17.349 20.100 15.330 18.708 21.703 19.975 18.735  0.000
```

# Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
  - Algoritmo de K-medias.
  - Una variante robusta: PAM
- 4 Técnicas jerárquicas.
  - Métodos aglomerativos.
  - Métodos divisivos.
- 5 Spectral Clustering
- 6 Comparación de particiones

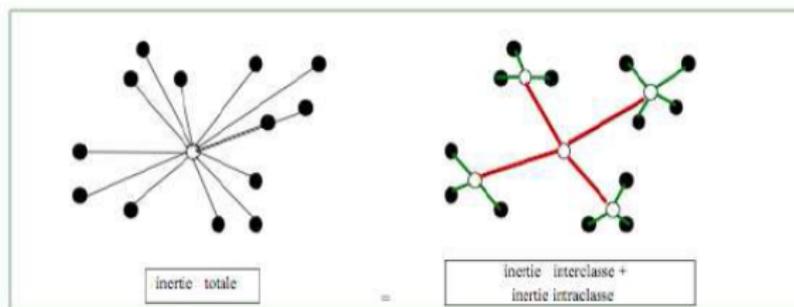
# Descomposición de la inercia

Si  $\mu = \frac{1}{n} \sum_i x_i = \bar{x}$ , se puede descomponer la suma de cuadrados de la siguiente manera:

$$\begin{aligned} \sum_{i=1}^n (x_i - \bar{x})^2 &= \sum_{k=1}^K \sum_{i=1}^n z_{ik} (x_i - \bar{x}_k)^2 + \sum_{k=1}^K n_k (\bar{x}_k - \bar{x})^2 \\ &= \sum_{k=1}^K \sum_{i=1}^{n_k} (x_{i(k)} - \bar{x}_k)^2 + \sum_{k=1}^K n_k (\bar{x}_k - \bar{x})^2 \end{aligned}$$

donde  $z_{ik} = \begin{cases} 1 & \text{si } x_i \in \text{cluster } C_k \\ 0 & \text{si } x_i \notin \text{cluster } C_k \end{cases}$

Observar que es parecido a  $\text{Var}[X] = E[\text{Var}[X | K]] + \text{Var}[E[X | K]]$



La cuenta anterior se obtiene de:

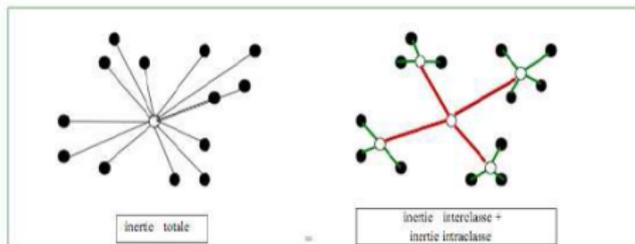
$$\begin{aligned}\sum_i (x_i - \bar{x})^2 &= \sum_i \sum_k z_{ik} (x_i - \bar{x})^2 \\ &= \sum_i \sum_k z_{ik} (x_i - \bar{x}_k + \bar{x}_k - \bar{x})^2 \\ &= \sum_k \sum_i z_{ik} ((x_i - \bar{x}_k)^2 + 2(x_i - \bar{x}_k)(\bar{x}_k - \bar{x}) + (\bar{x}_k - \bar{x})^2) \\ &= \sum_k \sum_i z_{ik} (x_i - \bar{x}_k)^2 + 2 \sum_k \sum_i z_{ik} (x_i - \bar{x}_k)(\bar{x}_k - \bar{x}) + \sum_k \sum_i z_{ik} (\bar{x}_k - \bar{x})^2 \\ &= \sum_k \sum_i z_{ik} (x_i - \bar{x}_k)^2 + 2 \sum_k (\bar{x}_k - \bar{x}) \left( \sum_i z_{ik} x_i - \sum_i z_{ik} \bar{x}_k \right) + \sum_k n_k (\bar{x}_k - \bar{x})^2 \\ &= \sum_k \sum_i z_{ik} (x_i - \bar{x}_k)^2 + 2 \sum_k (\bar{x}_k - \bar{x}) (n_k \bar{x}_k - n_k \bar{x}_k) + \sum_k n_k (\bar{x}_k - \bar{x})^2 \\ &= \sum_k \sum_i z_{ik} (x_i - \bar{x}_k)^2 + \sum_k n_k (\bar{x}_k - \bar{x})^2.\end{aligned}$$

# Técnicas de particionamiento

- Estas técnicas buscan particionar el conjunto de datos en un número especificado de grupos  $K$ ,  $1 \leq K \leq n$ , minimizando algún criterio o función objetiva que indica la “bondad” (en términos del objetivo del clustering) de cada partición.
- El enfoque más utilizado consiste en buscar la partición  $\mathcal{C} = \{C_1, \dots, C_K\}$  de  $n$  individuos en  $K$  grupos de forma tal de minimizar la suma de las distancias respecto al *centroide* del grupo,  $\mu_k$ :

$$SCD(\mathcal{C}) = \sum_{k=1}^K \sum_{x_i \in C_k} d^2(x_i, \mu_k)$$

- Cuando la distancia utilizada es la euclídea y el centroide es el promedio de las observaciones del grupo, éste es el criterio de *mínima suma de cuadrados* dentro de los grupos, o sea  $SCD(\mathcal{C}) = W(\mathcal{C})$ .
- En este caso, ya que  $T$  es constante, minimizar  $W$  es equivalente en maximizar  $B$ .



- El problema parece entonces relativamente simple: considerar todas las particiones posibles de  $n$  individuos en  $K$  grupos y seleccionar aquella con el valor más bajo de  $SCD(\mathcal{C})$ .
- Desafortunadamente, en la práctica esta solución no es viable. El número total de particiones a considerar,  $S(n, K)$  (equivalente a buscar la cantidad de funciones sobreyectivas de un conjunto de  $n$  elementos en otro de  $K$  elementos), es por lo general tan elevado que la enumeración completa es imposible incluso para la computadora más rápida!
- Se puede demostrar que:

$$S(n, K) = \frac{1}{K!} \sum_{k=1}^K (-1)^{K-k} \binom{K}{k} k^n$$

(Hastie y otros, 2009).

$n$	$K$	$S(n, K)$
15	3	2375101
20	4	$\approx 4.52 \times 10^{10}$
25	8	$\approx 6.9 \times 10^{17}$
100	5	$\approx 6.57 \times 10^{67}$

**Cuadro:** Número de posibles particiones según la cantidad de observaciones  $n$  y grupos  $K$ .

- Debido a este problema, los distintos algoritmos que se han desarrollado solo son capaces de examinar una fracción (a menudo muy) menor de las posibles particiones.

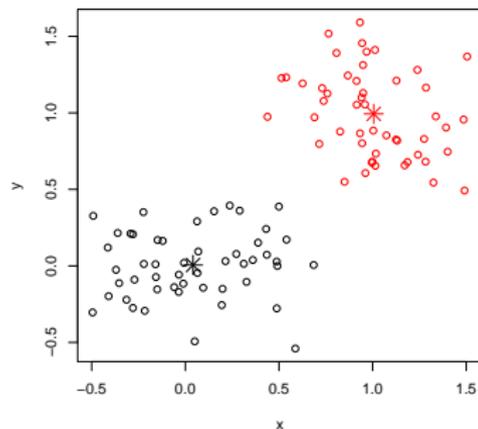
- El algoritmo de  $K$ -medias es un algoritmo particional y fue propuesto en los '50 (Jain, 2009)
- A pesar de que su primera aparición es desde hace más de 50 años sigue siendo de los algoritmos más utilizados para clustering por su facilidad de implementación, simpleza y buenos resultados empíricos.

- **Inicialización** : Se elige aleatoriamente  $K$  puntos del espacio (serán los centros de los grupos) o una partición inicial aleatoria con  $K$  clusters (son equivalentes).
- Se iteran las dos etapas siguientes hasta que se cumpla que el criterio de minimización de la inercia intraclases se estabilice o hasta una cantidad de iteraciones fijas:
  - 1 Todos los individuos están asignados a una clase cuyo centro es el más cercano (respecto de la distancia elegida). Se construyen entonces  $K$  clases.
  - 2 Se calculan los baricentros (centroides) de las clases creadas que son los nuevos centros que se consideraran en la etapa siguiente.

Figura: Algoritmo K-means

El algoritmo  $K$ -medias requiere del usuario los siguientes parámetros: número de clusters, inicialización de los clusters (centros), distancia (en general la distancia euclídea, es la distancia que usa R).

```
x <- rbind(matrix(rnorm(100, sd = 0.3), ncol = 2),  
matrix(rnorm(100, mean = 1, sd = 0.3), ncol = 2))  
colnames(x) <- c("x", "y")  
(cl <- kmeans(x, 2, iter.max=2))  
plot(x, col = cl$cluster)  
points(cl$centers, col = 1:2, pch = 8, cex = 2)
```



# Grupos obtenidos en la iteración 1 de K-medias

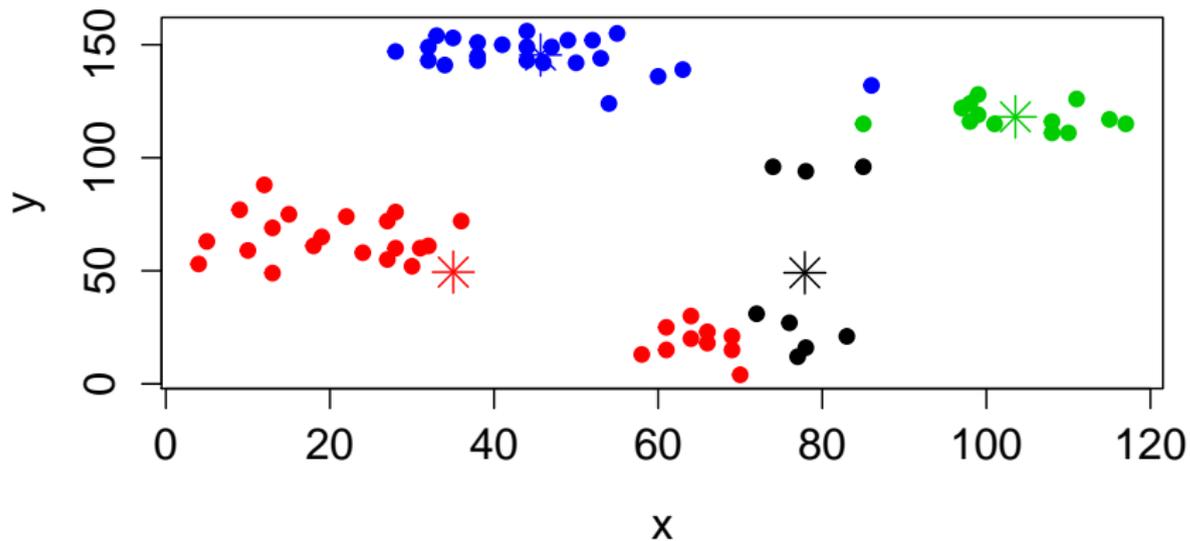


Figura: Grupos obtenidos en la iteración 1 de K-medias.

# Grupos obtenidos en la iteración 2 de K-medias

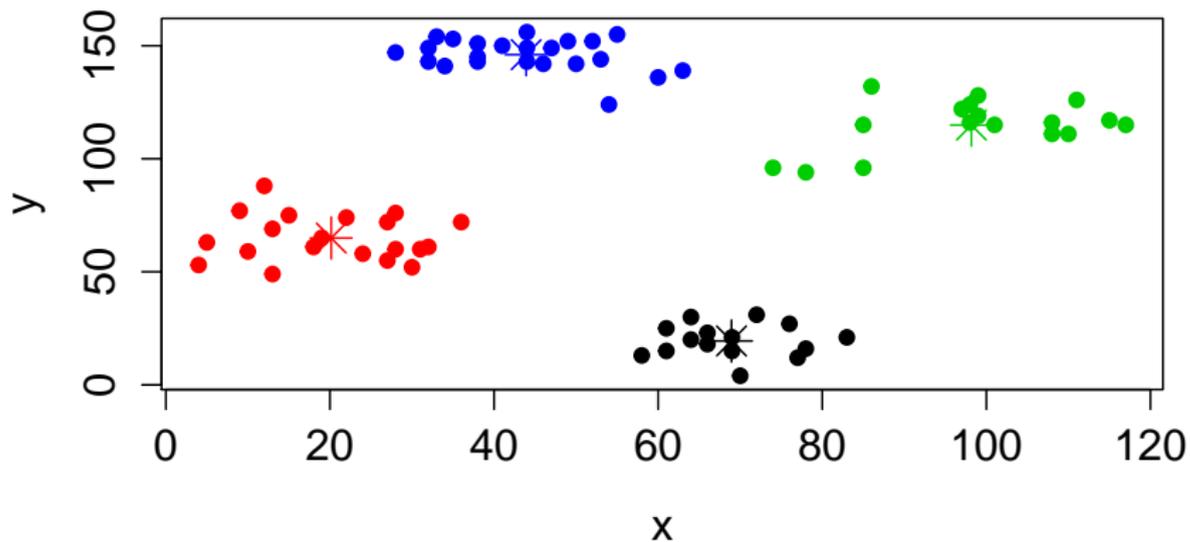


Figura: Grupos obtenidos en la iteración 2 de K-medias.

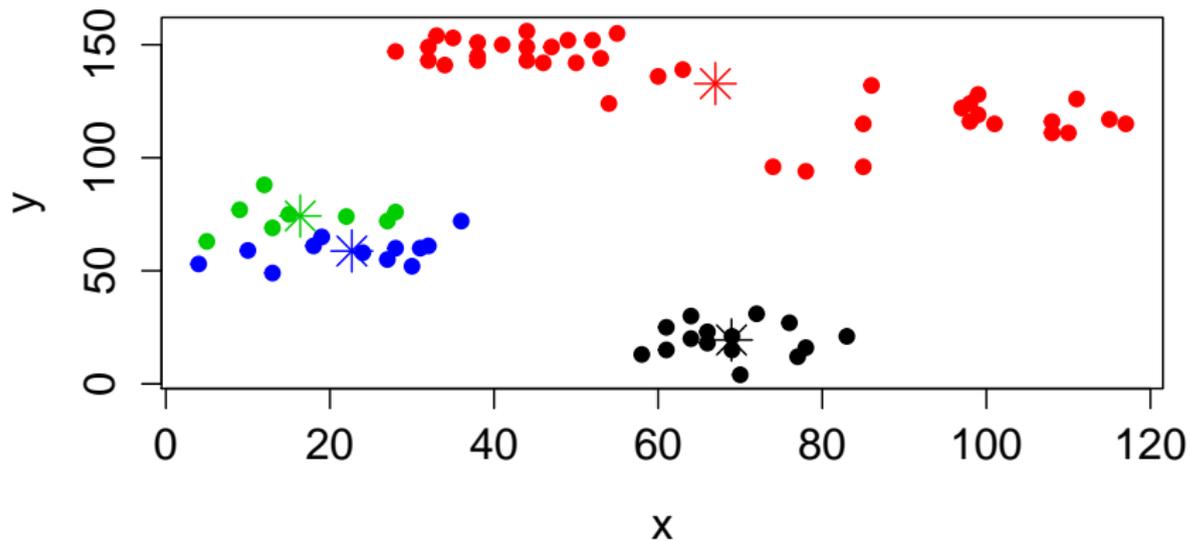


Figura: Grupos obtenidos en la iteración 10 de K-medias. Los grupos se estabilizan en un óptimo local.

Given an initial set of  $K$ -means  $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_K$ , the algorithm proceeds by alternating between two steps:

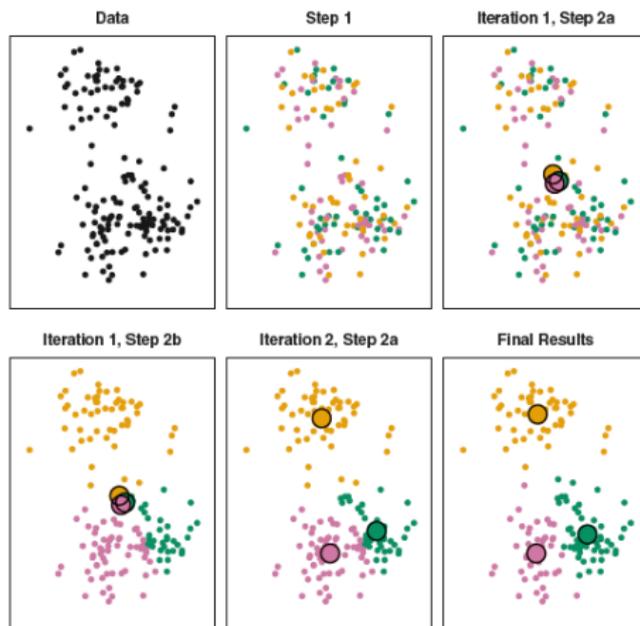
- 1 **Assignment step** Assign each observation to the cluster whose mean has the least squared euclidean distance], this is intuitively the "nearest" mean. Mathematically, this means partitioning the observations according to the groups generated by the means.

$$C_k^{(t)} = \{\mathbf{x}_i : \|\mathbf{x}_i - \bar{\mathbf{x}}_k^{(t)}\|^2 \leq \|\mathbf{x}_i - \bar{\mathbf{x}}_{k'}^{(t)}\|^2 \forall k', 1 \leq k' \leq k\}$$

- 2 **Update step** Calculate the new means to be the centroids of the observations in the new clusters.  $\bar{\mathbf{x}}_k^{(t+1)} = \frac{1}{|C_k^{(t)}|} \sum_{\mathbf{x}_i \in C_k^{(t)}} \mathbf{x}_i$

The algorithm has converged when the assignments no longer change. There is no guarantee that the optimum is found using this algorithm.

The method divides the space by several hyperplanes

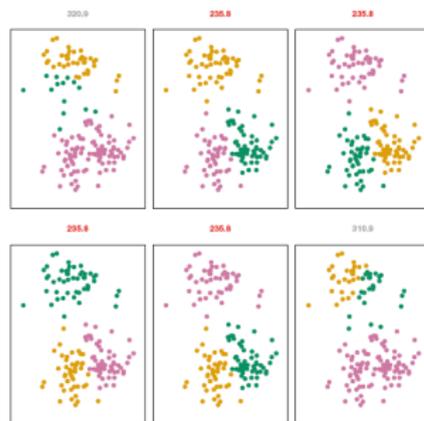


**FIGURE 10.6.** The progress of the  $K$ -means algorithm on the example of Figure 10.5 with  $K=3$ . Top left: the observations are shown. Top center: in Step 1 of the algorithm, each observation is randomly assigned to a cluster. Top right: in Step 2(a), the cluster centroids are computed. These are shown as large colored disks. Initially the centroids are almost completely overlapping because the initial cluster assignments were chosen at random. Bottom left: in Step 2(b), each observation is assigned to the nearest centroid. Bottom center: Step 2(a) is once again performed, leading to new cluster centroids. Bottom right: the results obtained after ten iterations.

- A cada iteración el criterio  $J_w = \sum_{k=1}^K \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \bar{\mathbf{x}}_k\|^2$  disminuye.
- El algoritmo converge rápidamente al menos hacia un mínimo local de  $J_w$ .
- Algunos inconvenientes de K-medias y los métodos particionales:
  - las variables son cuantitativas (por lo general).
  - hay que elegir el número de clusters al principio
  - son sensibles a la inicialización: inicializaciones distintas pueden dar clusters distintos .
- Algunas soluciones:
  - para el número de clusters: visualización, consideraciones sobre el problema, criterios de penalización (penalizan la cantidad de clusters, podemos citar por ejemplos AIC o BIC)
  - inicialización: como el algoritmo puede converger a mínimos locales, una solución es correr el algoritmo con diferentes particiones iniciales aleatorias y elegir el resultado con menor función objetivo.  
Otra posibilidad es reagrupar las observaciones  $x_i$  que están siempre en el mismo cluster.

## Multiples inicializaciones.

Because the K-means algorithm finds a local rather than a global optimum, the results obtained will depend on the initial (random) cluster assignment of each observation in Step 1 of Algorithm 10.1. For this reason, it is important to run the algorithm multiple times from different random initial configurations. Then one selects the best solution, i.e. that for which the objective (10.11) is smallest. Figure 10.7 shows the local optima obtained by running K-means clustering six times using six different initial cluster assignments, using the toy data from Figure 10.5. In this case, the best clustering is the one with an objective value of 235.8.



**FIGURE 10.7.** K-means clustering performed six times on the data from Figure 10.5 with  $K = 3$ , each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of which resulted in a smaller value of the objective and provides better separation between the clusters. Those labeled in red all achieved the same best solution, with an objective value of 235.8.

Buscar un codo en la gráfica del criterio (within groups sum of squares)

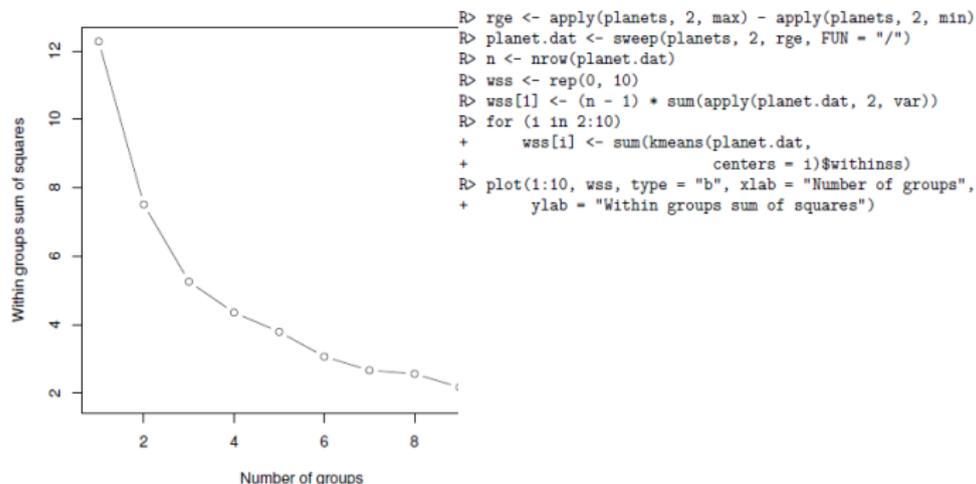


Figure 18.7 Within-cluster sum of squares for different number of groups for the exoplanet data.

- Al igual que K-medias, el algoritmo de *partitioning around medoids* (PAM) necesita una configuración (partición) inicial y un número preespecificado de grupos.
- Por otro lado, PAM busca los K “individuos representativos” (o *medoides*) entre el conjunto de observaciones (mientras que K-medias utiliza los promedios del grupo), que minimizan la suma de las disimilaridades al resto de los integrantes. No se necesita de la distancia euclídea.
- En general PAM es más robusto que K-medias y requiere como argumento de entrada solamente la matriz de disimilaridades entre observaciones y no los datos originales.
- Como contrapartida es más intensivo computacionalmente, debido principalmente a la búsqueda de *medoides* (Izenman, 2008).

# Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
  - Algoritmo de K-medias.
  - Una variante robusta: PAM
- 4 **Técnicas jerárquicas.**
  - Métodos aglomerativos.
  - Métodos divisivos.
- 5 Spectral Clustering
- 6 Comparación de particiones

- Los resultados de aplicar las técnicas de particionamiento (K-medias o PAM por ejemplo), dependen de la elección del número de clusters y de una configuración inicial.
- En cambio, los métodos de *clustering jerárquico* no requieren estas especificaciones.
- En su lugar, estas técnicas necesitan que el usuario especifique una *medida de disimilaridad entre grupos* (disjuntos), basada en las disimilaridades entre las observaciones de los grupos.
- Como su nombre sugiere, estas técnicas producen representaciones jerárquicas en las cuales los clusters en cada nivel de la jerarquía son creados uniendo clusters del siguiente nivel inferior (y por ende podemos reconstruir la “historia”).
- En el nivel más bajo cada cluster contiene una única observación y en el más alto hay un sólo cluster con todas las observaciones.
- Las estrategias para el clustering jerárquicos se dividen en dos paradigmas básicos: *aglomerativos* y *divisivos*.
- Son computacionalmente más complejos que K-means (típicamente  $O(n^2 \log(n))$  vs.  $O(n)$ ) y conviene aplicarlos en muestras chicas.

Las medidas de disimilaridad más comunes entre clusters son:

- *single linkage* o vecino más cercano:

$$d_{\min}(C_1, C_2) = \min_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$

- *complete linkage* o vecino más lejano:

$$d_{\max}(C_1, C_2) = \max_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$

- *average linkage*:

$$d_{\text{media}}(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{x_i \in C_1, x_j \in C_2} d(x_i, x_j)$$

- *método centroide*:

$$d_{\text{cent}}(C_1, C_2) = d(\mu_1, \mu_2)$$

donde  $\mu_1$  y  $\mu_2$  son los centroides de  $C_1$  y  $C_2$  respectivamente

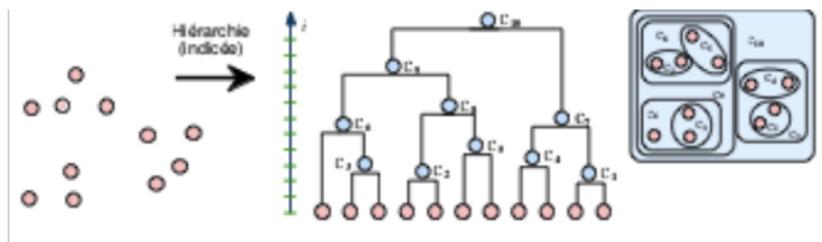
- *de Ward*:

$$d_{\text{ward}}(C_1, C_2) = \sqrt{\frac{|C_1||C_2|}{|C_1| + |C_2|}} d(\mu_1, \mu_2)$$

- Ninguna de estas medidas es *uniformemente* mejor que las otras para todos los problemas de clustering (Izenman, 2008).
- El vecino más cercano a menudo produce largas “cadenas” de clusters, unidas por unos pocos puntos cercanos (lo cual podrá no ser deseado), mientras que el vecino más lejano tiende a producir muchos clusters, pequeños y compactos.
- La distancia promedio es dependiente del tamaño de los clusters, a diferencia de las otras dos.
- Si los datos presentan una estructura de grupos bien diferenciados, entonces los cuatro métodos tenderán a dar los mismos resultados.

# Métodos aglomerativos

- Estos métodos comienzan con cada observación representando un solo cluster.
- En cada uno de los siguientes pasos los dos cluster más cercanos (menos disímiles) son unidos en un único grupo, produciendo un cluster menos que en el nivel inmediato anterior.
- El resultado es un árbol (dendograma) donde las hojas son las muestras originales y representa a las fusiones sucesivas.
- Por lo tanto, es necesario definir una medida de disimilaridad entre clusters o grupos de observaciones.
- Distintas medidas dan lugar a variantes que pueden producir resultados diferentes.



- Los algoritmos de clustering divisivos comienzan con un único cluster con todas las observaciones y recursivamente dividen uno de los clusters existentes en dos clusters “hijos” hasta obtener tantos grupos como observaciones.
- Dividir un cluster es computacionalmente más demandante que unir dos, dado que no solo se debe encontrar el cluster a ser dividido sino que también las observaciones que formarn los dos nuevos grupos deben ser identificadas.
- Por este motivo, los métodos divisivos son menos utilizados en la práctica (Varmuza y Filmoser, 2009).
- Los resultados del clustering jerárquico suelen ser representados mediante un diagrama de árbol jerárquico, conocido como *dendograma*.
- Grupos u observaciones que son más similares son combinados a bajas alturas, mientras que los más dismiles lo hacen a alturas grandes.
- Una partición de los datos en un número dado de grupos puede obtenerse “cortando” el dendograma en un nivel apropiado de altura.

# Ejemplo numérico en R

```
X <- rbind(c(1, 3), c(2, 4), c(1, 5), c(5, 5), c(5, 7), c(4, 9), c(2, 8), c(3, 10))
> X
      [,1] [,2]
[1,]    1    3
[2,]    2    4
[3,]    1    5
[4,]    5    5
[5,]    5    7
[6,]    4    9
[7,]    2    8
[8,]    3   10
> (D <- dist(X))
      1      2      3      4      5      6      7
2 1.414214
3 2.000000 1.414214
4 4.472136 3.162278 4.000000
5 5.656854 4.242641 4.472136 2.000000
6 6.708204 5.385165 5.000000 4.123106 2.236068
7 5.099020 4.000000 3.162278 4.242641 3.162278 2.236068
8 7.280110 6.082763 5.385165 5.385165 3.605551 1.414214 2.236068

> par(mfrow = c(2, 2), mex = .8)
> plot(X, ylim = c(.8 * min(X[, 1]), 1.2 * max(X[, 2])), pch = 19, cex = 1.5,
+ xlab = expression(x[1]), ylab = expression(x[2]))
> text(X[, 1], 1.25 + X[, 2], 1:8, cex = 1.5)
> plot(hclust(dist(X), 'single'), main = 'Single linkage', xlab = '', cex = 1.5)
> plot(hclust(dist(X), 'complete'), main = 'Complete linkage', xlab = '', cex = 1.5)
> plot(hclust(dist(X), 'average'), main = 'Average linkage', xlab = '', cex = 1.5)
```

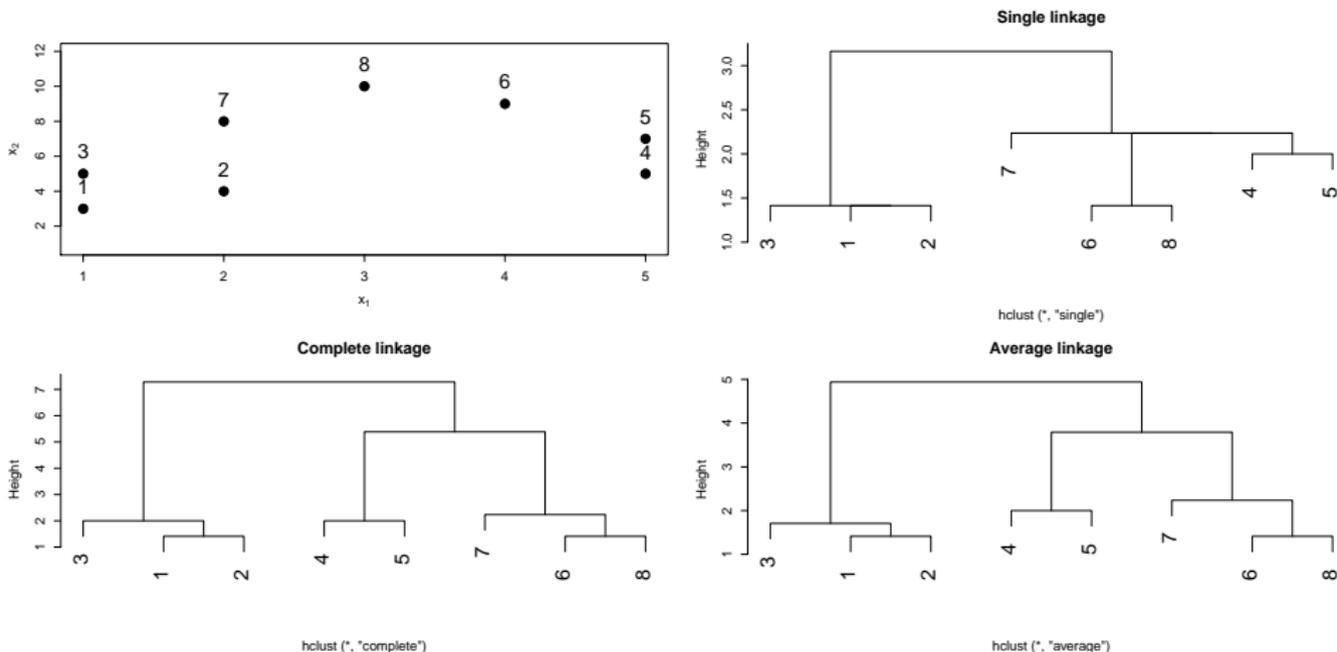


Figura: Pequeño ejemplo de clustering jerárquico aglomerativo.

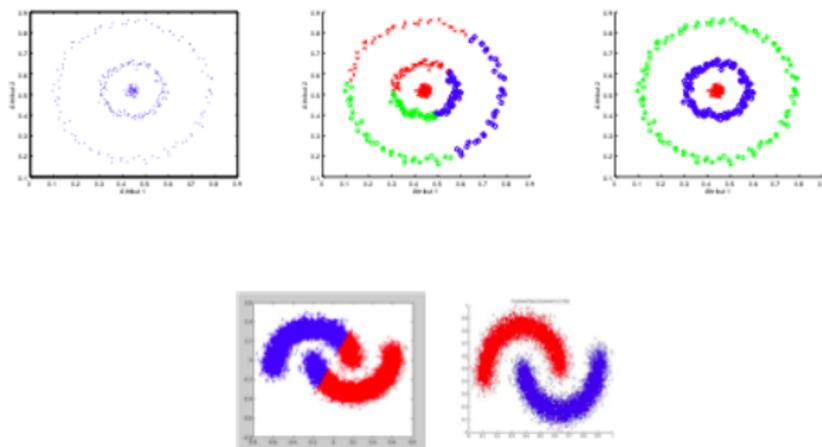
# Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
  - Algoritmo de K-medias.
  - Una variante robusta: PAM
- 4 Técnicas jerárquicas.
  - Métodos aglomerativos.
  - Métodos divisivos.
- 5 Spectral Clustering
- 6 Comparación de particiones

# Spectral Clustering

(A Tutorial on Spectral Clustering, U. Von Luxburg 2007 )

Traditional methods as  $k$ -means use elliptical metrics and may not detected non-convex clusters. Spectral Clustering converts a clustering problem into a graph partition problem and outperforms many traditional algorithms.



**Figura:** First row at left: the data, at center the clustering obtained by  $k$ -means and at right the clustering obtained by spectral clustering. Second row: Banana data set with  $k$ -means and with spectral clustering

## General ideas. Three steps

- 1 Construct a similarity graph, as a K-NN graph or an epsilon-graph or a similarity graph  $\left( s_{ij} = s(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \right)$ , from all the data points:

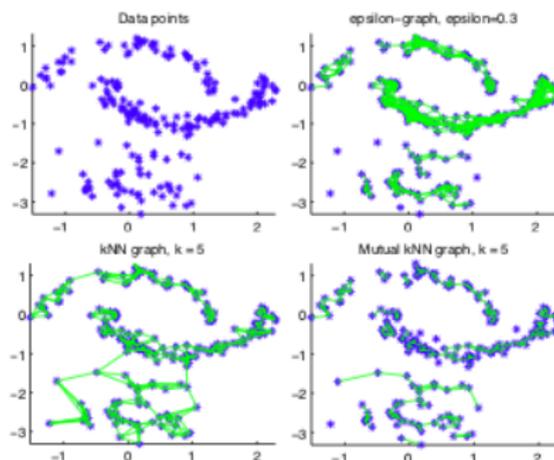


Figura: Different similarity graphs (Von Luxburg, 2007)

- 2 Embed the data points in a low dimensional space (*spectral embedding*) in which the cluster are more obvious, with the use of the eigenvectors of the graph Laplacian.
- 3 Apply a classical cluster algorithm such *k*-means in this new space to partition the embedding.

# Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
  - Algoritmo de K-medias.
  - Una variante robusta: PAM
- 4 Técnicas jerárquicas.
  - Métodos aglomerativos.
  - Métodos divisivos.
- 5 Spectral Clustering
- 6 Comparación de particiones

- Otro de los problemas que surge es el de validación de los clusters obtenidos. Por ejemplo el algoritmo de K-medias siempre encuentra clusters, no importa si los hay o no.
- Algunos criterios de validez de los clusters:
  - criterios internos: solo se basan en los datos y en el algoritmo usado. Una medida interna es la noción de estabilidad de los clusters, donde se mide la variabilidad de los clusters al aplicar el algoritmo a diferentes submuestras de los datos;
  - criterios relativos: se comparan diferentes estructuras, por ejemplo se hacen clusters con diferentes algoritmos y se elige cuales son mejores en algún sentido;
  - criterios externos: se valida en base a información a priori de etiquetas de los datos (aunque en el caso de existir etiquetas se podrán usar métodos supervisados).

Una forma intuitiva de comparar particiones es contando los pares de observaciones que son clasificados de la misma forma en ambas particiones.

El conjunto de todos los pares (no ordenados) de  $\mathcal{L}$  es la unión disjunta de los siguientes conjuntos:

- $S_a = \{\text{pares que están en el mismo cluster en } \mathcal{C} \text{ y } \mathcal{C}'\}$
- $S_b = \{\text{pares que están en distintos clusters en } \mathcal{C} \text{ y } \mathcal{C}'\}$
- $S_c = \{\text{pares que están en el mismo cluster en } \mathcal{C} \text{ y distinto cluster en } \mathcal{C}'\}$
- $S_d = \{\text{pares que están en distintos clusters en } \mathcal{C} \text{ y en el mismo cluster en } \mathcal{C}'\}$

Se denotan como  $a, b, c$  y  $d$  a los cardinales de  $S_a, S_b, S_c$  y  $S_d$ , respectivamente y es claro que:

- $S_a, S_b, S_c$  y  $S_d$  son disjuntos dos a dos
- $a + b + c + d = n(n-1)/2 = \binom{n}{2}$ .

Para un conjunto de datos con  $n = 8$  observaciones, se definen las particiones con 3 clases:

$$\mathcal{C} = \{C_1 = (x_1, x_2, x_3), C_2 = (x_4, x_5), C_3 = (x_6, x_7, x_8)\}$$

$$\mathcal{C}' = \{C'_1 = (x_6, x_7), C'_2 = (x_1, x_2, x_3, x_4, x_5), C'_3 = (x_8)\}$$

A partir de la comparación, dichos elementos se clasifican en los siguientes conjuntos definidos anteriormente:

- $S_a = \{(x_1, x_2), (x_1, x_3), (x_2, x_3), (x_4, x_5), (x_6, x_7)\}$
- $S_b = \{(x_1, x_6), (x_1, x_7), (x_1, x_8), (x_2, x_6), (x_2, x_7), (x_2, x_8), (x_3, x_6), (x_3, x_7), (x_3, x_8), (x_4, x_6), (x_4, x_7), (x_4, x_8), (x_5, x_6), (x_5, x_7), (x_5, x_8)\}$
- $S_c = \{(x_6, x_8), (x_7, x_8)\}$
- $S_d = \{(x_1, x_4), (x_2, x_4), (x_3, x_4), (x_1, x_5), (x_2, x_5), (x_3, x_5)\}$

Por lo cual el conteo de los  $n(n-1)/2 = 28$  pares tiene como resultado las cantidades  $a = 5$ ,  $b = 15$ ,  $c = 2$  y  $d = 6$ .

- 1 Índice de Rand:  $R(\mathcal{C}, \mathcal{C}') = \frac{a+b}{a+b+c+d} = \frac{2(a+b)}{n(n-1)}$
- 2 Índice de Rand Ajustado:  $R_{adj}(\mathcal{C}, \mathcal{C}') = \frac{a - \frac{((a+d)(a+c) / (a+b+c+d))}{2}}{\frac{(a+d)+(a+c)}{2} - \frac{(a+d)(a+c)}{a+b+c+d}}$
- 3 Índice de Fowlkes-Mallows:  $FM(\mathcal{C}, \mathcal{C}') = \frac{a}{\sqrt{(a+c)(a+d)}}$
- 4 Índice de Jaccard:  $J(\mathcal{C}, \mathcal{C}') = \frac{a}{a+c+d}$

Con los datos del ejemplo:

$$R(\mathcal{C}, \mathcal{C}') = 0,72 \quad R_{adj}(\mathcal{C}, \mathcal{C}') = 0,36 \quad FM(\mathcal{C}, \mathcal{C}') = 0,57 \quad J(\mathcal{C}, \mathcal{C}') = 0,38$$

- 1 Información mutua:  $NMI_1(\mathcal{C}, \mathcal{C}') = \frac{I(\mathcal{C}, \mathcal{C}')}{\sqrt{H(\mathcal{C})H(\mathcal{C}')}}}$
- 2 Mínimo Error de Clasificación:  $MCE(\mathcal{C}, \mathcal{C}') = \min_{\sigma \in S_n} \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{y_i \neq \sigma(\hat{y}_i)\}}$

Con los datos del ejemplos:

$$NMI_1(\mathcal{C}, \mathcal{C}') = 0,670 \quad MCE(\mathcal{C}, \mathcal{C}') = 0,375$$

```
library(clv)
data(iris)
iris.data <- iris[,1:4]

# cluster data
pam.mod <- pam(iris.data,3) # create three clusters
v.pred <- as.integer(pam.mod$clustering) # get cluster ids associated to given data
v.real <- as.integer(iris$Species) # get also real cluster ids

# compare true clustering with those given by the algorithm
# 1. optimal solution:

# use only once std.ext function
std <- std.ext(v.pred, v.real)
# to compute three indices based on std.ext result
rand1 <- clv.Rand(std)
jaccard1 <- clv.Jaccard(std)
folk.mall1 <- clv.Folkes.Mallows(std)
```