

Introducción a la Ciencia de Datos

Clustering

Mathias Bourel

IMERL - Facultad de Ingeniería, Universidad de la República, Uruguay

May 31, 2021

Plan

- 1 **Introducción**
- 2 Distancias
- 3 Técnicas de particionamiento.
- 4 Técnicas jerárquicas.
- 5 Otros métodos
- 6 Comparación de particiones

Introducción

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.

Introducción

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.

Introducción

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.

Introducción

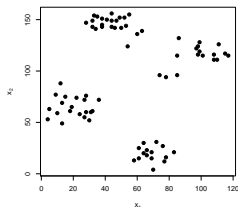


Figure: Ejemplo en dos dimensiones donde los grupos parecen “naturales”.

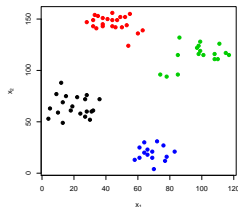


Figure: Grupos obtenidos mediante un algoritmo de clustering (PAM).

Introducción

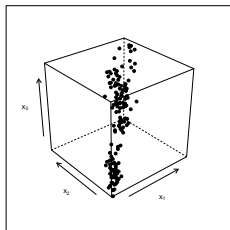


Figure: Ejemplo en tres dimensiones sobre los datos iris.

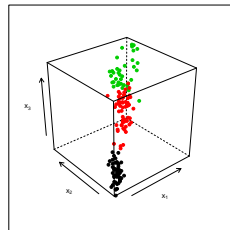


Figure: Grupos obtenidos mediante un algoritmo de clustering (PAM).

- Las grandes familias de técnicas son:
 - ▶ *particionamiento* (K-medias, PAM),
 - ▶ *jerárquicas* (aglomerativas, divisivas),
 - ▶ *basadas en modelos*,
 - ▶ *basadas en densidades*,
 - ▶ *basadas en grafos* (*Spectral Clustering*).
- Un aspecto central en todas ellas es la noción de similarad o de distancias entre los objetos a ser agrupados. Para ello es necesario en muchos casos utilizar medidas de distancia adecuadas entre los datos.

Plan

- 1 Introducción
- 2 Distancias**
- 3 Técnicas de particionamiento.
- 4 Técnicas jerárquicas.
- 5 Otros métodos
- 6 Comparación de particiones

Distancias.

- 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.

Distancias.

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

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

donde Σ 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) Por ejemplo si
 - a =cantidad de variables con respuesta 1 en ambos individuos
 - b =cantidad de variables con respuesta 0 en individuo i y 1 en individuo j .
 - c =cantidad de variables con respuesta 1 en individuo i y 0 en individuo j .
 - d =cantidad de variables con respuesta 0 en ambos individuos.

Observar que $a + b + c + d$ es la cantidad de variables p . El coeficiente de similitud de Sokal y Michener se define como:

$$s_{ij} = \frac{a + d}{p}$$

e induce una distancia entre individuos.

- 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.

Plan

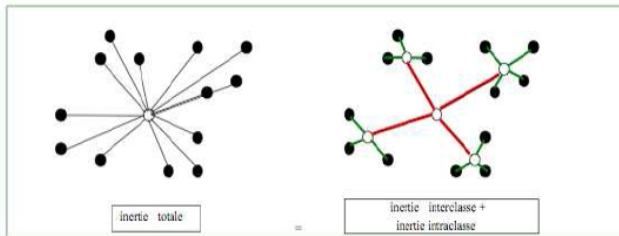
- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
 - Generalidades: descomposición de la inercia
 - Algoritmo de K-means.
 - Una variante robusta: PAM
- 4 Técnicas jerárquicas.
- 5 Otros métodos
- 6 Comparación de particiones

Descomposición de la inercia

Si $\bar{x} = \frac{1}{n} \sum_i x_i$, 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 $z_{ik} = \begin{cases} 1 & \text{si } x_i \in \text{cluster } C_k \\ 0 & \text{si } x_i \notin \text{cluster } C_k \end{cases}$



Descomposición de la inercia

La cuenta anterior se obtiene de:

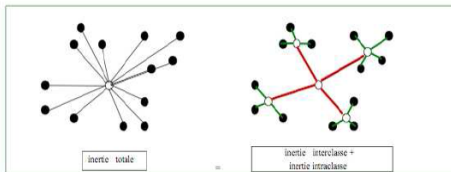
$$\begin{aligned}\sum_i (x_i - \bar{x})^2 &= \sum_i \sum_k z_{ik} (x_i - \bar{x})^2 \\ &= \sum_i \sum_k z_{ik} (x_i - \bar{x}_k + \bar{x}_k - \bar{x})^2 \\ &= \sum_k \sum_i z_{ik} \left((x_i - \bar{x}_k)^2 + 2(x_i - \bar{x}_k)(\bar{x}_k - \bar{x}) + (\bar{x}_k - \bar{x})^2 \right) \\ &= \sum_k \sum_i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

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, μ_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 euclidea 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 .



Técnicas de particionamiento

- 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 $SCD(C)$.
- 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	$\approx 6.57 \times 10^{67}$

Table: Número de posibles particiones según la cantidad de observaciones n y grupos K .

Técnicas de particionamiento: K-medias

- 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.

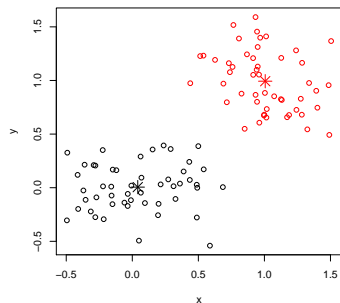
- Inicialización : Se elige aleatoriamente K puntos del espacio (serán los centros de los grupos) o una partición inicial aleatoria con K clusters (son equivalentes).
- Se iteran las dos etapas siguientes hasta que se cumpla que el criterio de minimización de la inercia intraclases se estabilice o hasta 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.

Figure: 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).

Técnicas de particionamiento: K-medias

```
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

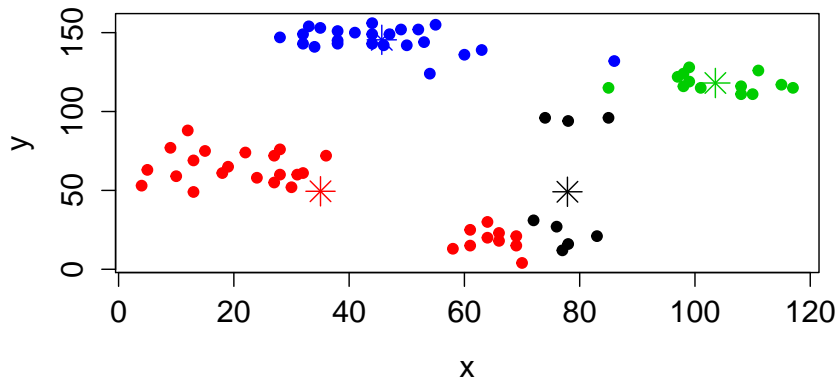


Figure: Grupos obtenidos en la iteración 1 de K-medias.

Grupos obtenidos en la iteración 2 de K-medias

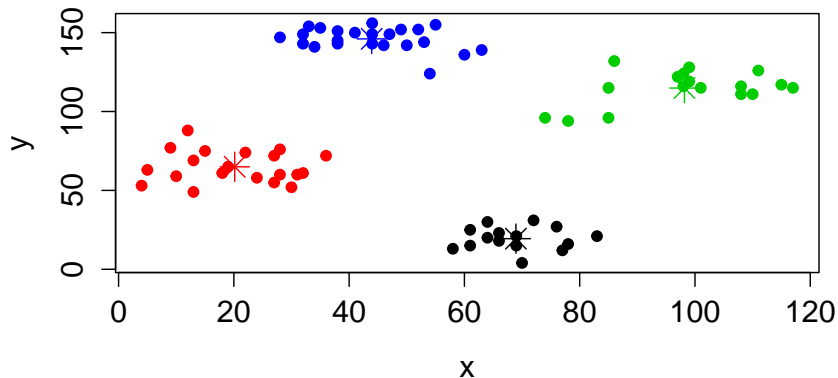


Figure: Grupos obtenidos en la iteración 2 de K-medias.

Grupos obtenidos en la iteración 10 de K-medias

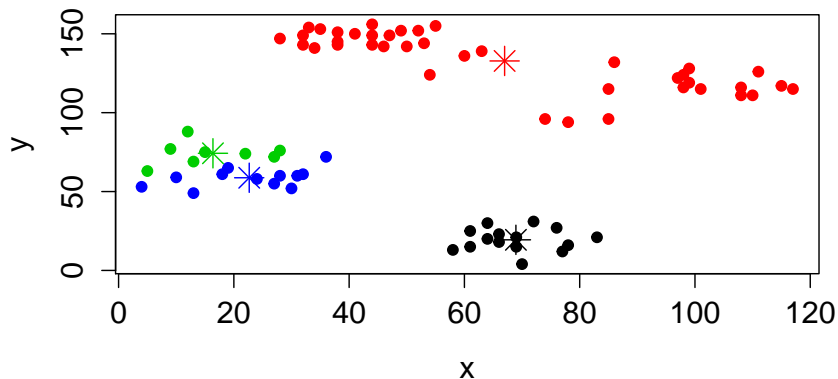


Figure: Grupos obtenidos en la iteración 10 de K-medias. Los grupos se estabilizan en un óptimo local.

Ejemplo tomado del ISLR

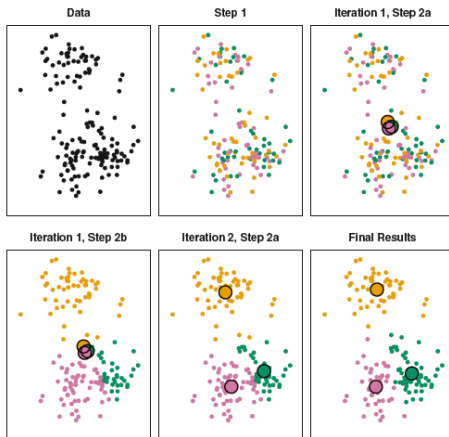


FIGURE 10.6. The progress of the K-means algorithm on the example of Figure 10.5 with $K=3$. Top left: the observations are shown. Top center: in Step 1 of the algorithm, each observation is randomly assigned to a cluster. Top right: in Step 2(a), the cluster centroids are computed. These are shown as 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 la función objetivo $\sum_{k=1}^K \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \bar{\mathbf{x}}_k\|^2$ disminuye.
- El algoritmo converge rápidamente al menos hacia un mínimo local (por eso es importante aplicarlos varias veces para decidir la partición)
- 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 .

K-means: varias inicializaciones

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.

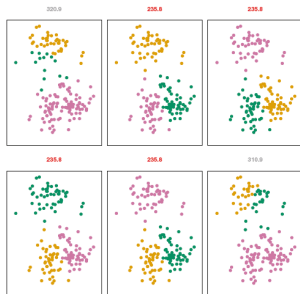


FIGURE 10.7. K-means clustering performed six times on the data from Figure 10.5 with $K = 3$, each time with a different random assignment of the observations in Step 1 of the K-means algorithm. Above each plot is the value of the objective (10.11). Three different local optima were obtained, one of 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.

K-means: Criterio de parada

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

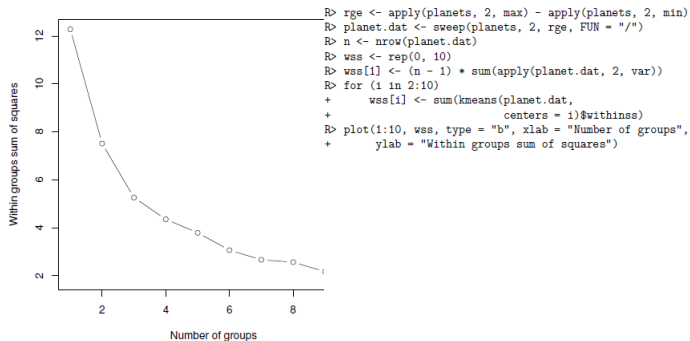


Figure 18.7 Within-cluster sum of squares for different number of the exoplanet data.

Una variante robusta: PAM

- 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.
- 4 **Técnicas jerárquicas.**
 - Métodos aglomerativos.
 - Métodos divisivos.
- 5 Otros métodos
- 6 Comparación de particiones

Técnicas jerárquicas

- 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.

Medidas de disimilaridad

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*:

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

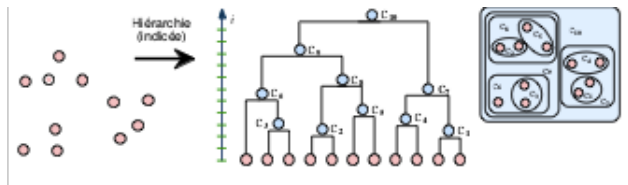
donde μ_1 y μ_2 son los centroides de C_1 y C_2 respectivamente

- *de Ward*:

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

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 formarán los dos nuevos grupos deben ser identificadas.
- Por este motivo, los métodos divisivos son menos utilizados en la práctica.

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)
```


Ejemplo en R

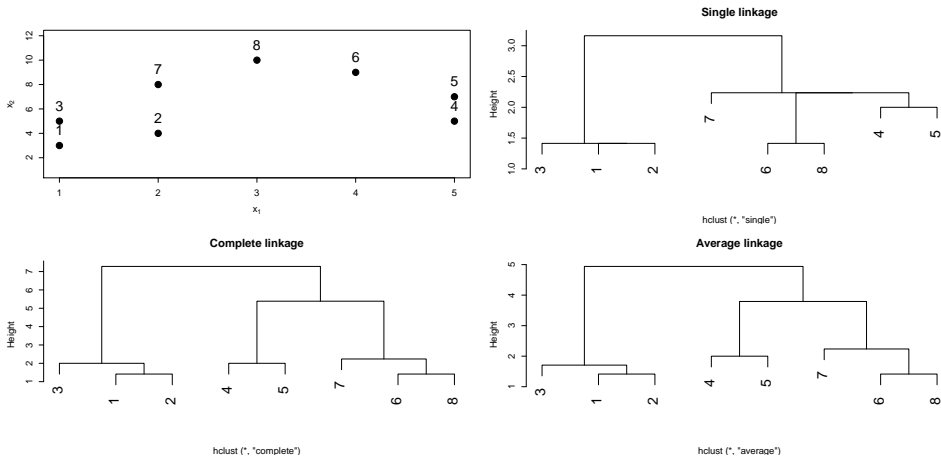


Figure: Pequeño ejemplo de clustering jerárquico aglomerativo.

Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
- 4 Técnicas jerárquicas.
- 5 Otros métodos**
 - Métodos basados en modelos
 - Métodos basados en densidad
 - Métodos basados en grafos
- 6 Comparación de particiones

Clustering basado en modelos

- En las técnicas de clustering presentadas anteriormente no se hace ningún supuesto acerca de la distribución de los datos (excepto que se asume una estructura de grupos).
- En cambio, en las técnicas de clustering *basado en modelos* se asume un modelo estadístico para los clusters, donde cada uno de ellos puede ser representado por una distribución de probabilidad multivariada.
- La distribución conjunta de los datos se dice que es una *mezcla* de distribuciones con componentes dadas por cada uno de los clusters:

$$f(x|\theta) = \sum_{j=1}^K \pi_j f(x|\theta_j), \quad \pi_j \geq 0, \quad \sum_{j=1}^K \pi_j = 1$$

- Luego, el algoritmo debe buscar la estimación de los parámetros que “mejor” se adapten a los datos y al modelo propuesto (*máxima verosimilitud*), así como el grado de pertenencia de cada observación a los distintos grupos.
- Para ello se utiliza el algoritmo Expectation - Maximization (EM).

Clustering basados en densidad: DBSCAN

DBSCAN: Density-Based Spatial Clustering of Applications with Noise.

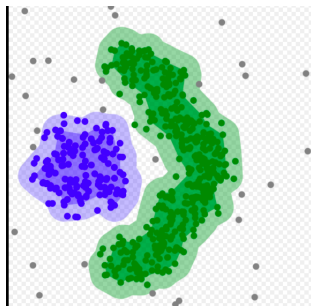


Figure: DBSCAN puede encontrar clusters que no son linealmente separables. Este conjunto de datos no puede ser correctamente agrupado con k-means o Gaussian Mixture EM clustering. (Wikipedia)

DBSCAN (Martin Ester et al., 1996) es un algoritmo de agrupamiento basado en densidad que funciona bajo el supuesto de que los agrupamientos son regiones densas en el espacio separados por regiones de menor densidad.

No se necesita especificar K.

Clustering basados en grafos: 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.

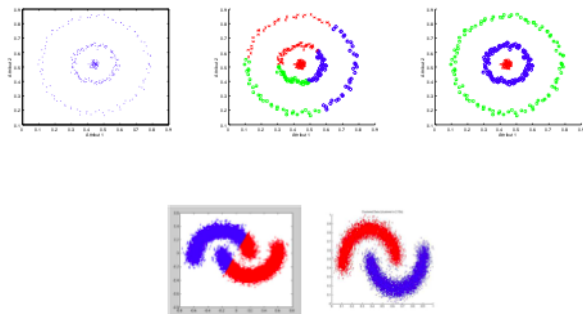


Figure: First row at left: the data, at center the clustering obtained by k -means and at right the clustering obtained by spectral clustering. Second row: Banana data set with k -means and with spectral clustering

Plan

- 1 Introducción
- 2 Distancias
- 3 Técnicas de particionamiento.
- 4 Técnicas jerárquicas.
- 5 Otros métodos
- 6 Comparación de particiones**

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).

Comparación de particiones

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:

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

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
- $a + b + c + d = n(n - 1)/2 = \binom{n}{2}$.

Comparación de particiones

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 $d = 6$.

Algunos índices de comparación

- 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: $R_{adj}(\mathcal{C}, \mathcal{C}') = \frac{a - \frac{(a+d)(a+c)}{(a+b+c+d)}}{\frac{(a+d)+(a+c)}{2} - \frac{(a+d)(a+c)}{a+b+c+d}}$
- 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 de los ejemplos:

$$NMI_1(\mathcal{C}, \mathcal{C}') = 0.670 \quad MCE(\mathcal{C}, \mathcal{C}') = 0.375$$

Ejemplo

```
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$clustering) # get cluster ids associated to gi
v.real <- as.integer(iris$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)
```