Técnicas de Agrupamiento

Minería de Datos

Máster en Análisis Masivo de Datos

José T. Palma

Departamento de Ingeniería de la Información y las Comunicaciones
Universidad de Murcia

2025

UNIVERSIDAD DE MURCIA

Contenidos de la presentación

Introducción

- En el caso del aprendizaje o supervisado no tenemos ninguna información acerca de la organización de los elementos en grupos o clases.
- Por lo tanto, el objetivo consiste en encontrar dicha organización en base a la relación entre los elementos.
- No existe información previa sobre dicha organización y la interpretación de las clases y los grupos obtenidos hay que hacerla a posteriori.
- Para ello podemos aplicar técnicas de agrupamiento o "clustering":
  - Identificar distintos grupos de clientes en un banco para personalizar las ofertas de productos financieros.
  - Identificar distintos subgrupos en un tipo determinado de cáncer para ajustar los tratamientos.

Introducción

- La idea básica consiste en crear grupos que contengan elementos parecidos entre sí y que elementos dispares se coloquen en grupos diferentes.
- Una técnica de clustering es capaz de describir la estructura subyacente a un conjunto de datos analizando las similitudes y diferencias (p. e., distancias) entre los elementos del conjunto.
- El objetivo final es obtener un conjunto de clases o grupos:
  - Cuando estos grupos son disjuntos y cubren todo el conjunto de elementos se dice que el agrupamiento es "particional".
  - En algunos casos lo que interesa es una jerarquía de agrupamientos particionales anidados. En este caso tenemos un agrupamiento jerárquico que se suele representar mediante un dendograma.

Distancia y similaridad

- El concepto de similaridad y distancia es clave en las técnicas de agrupamiento ya que definen la lente que le vamos a dar al ordenador para que busque la estructura de los datos.
- Supongamos que tenemos n elementos (instancias u objetos) recogidos en un conjunto $\Omega = \{\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}, ..., \mathbf{e_n}\}$

Definición (Distancia)

Una medida de distancia sobre el conjunto Ω es una función d tal que:

$$\begin{array}{rccc} d: & \Omega \times \Omega & \rightarrow & \mathbb{R} \\ & d(\mathbf{e_i}, \mathbf{e_j}) & \rightarrow & d_{ij} \end{array}$$

Distancia y Similaridad

Distancia: propiedades

$\forall\ \mathbf{e_i}, \mathbf{e_j} \in \Omega$

1. $d(\mathbf{e_i}, \mathbf{e_j}) \geq 0$
2. $d(\mathbf{e_i}, \mathbf{e_i}) = 0$
3. $d(\mathbf{e_i}, \mathbf{e_j}) = d(\mathbf{e_j}, \mathbf{e_i})$

- Cuando además se cumple la propiedad de desigualdad triangular:

$$d(\mathbf{e_i}, \mathbf{e_j}) \leq d(\mathbf{e_i}, \mathbf{e_k}) + d(\mathbf{e_k}, \mathbf{e_j})\ \forall \mathbf{e_i}, \mathbf{e_j}, \mathbf{e_k} \in \Omega$$

diremos que la distancia es métrica y (Ω, d) forma un espacio métrico.

Distancia y similaridad

Definición (Similaridad)

Una medida de similaridad sobre el conjunto Ω es una función s tal que:

$$\begin{array}{rccc} s: & \Omega \times \Omega & \rightarrow & \mathbb{R} \\ & s(\mathbf{e_i}, \mathbf{e_j}) & \rightarrow & s_{ij} \end{array}$$

tal que $\forall\ \mathbf{e_i}, \mathbf{e_j} \in \Omega$:

1. $s(\mathbf{e_i}, \mathbf{e_i}) \in [0, 1]$
2. $1 = s(\mathbf{e_i}, \mathbf{e_i}) \geq s(\mathbf{e_i}, \mathbf{e_j})$
3. $s(\mathbf{e_i}, \mathbf{e_j}) = s(\mathbf{e_j}, \mathbf{e_i})$

Distancia y similaridad

- Un ejemplo de medida de similaridad puede ser la *similaridad del coseno*:

$$s(\mathbf{e_i}, \mathbf{e_j}) = \cos\theta = \frac{\mathbf{e_i}^T \mathbf{e_j}}{||\mathbf{e_i}||_2 ||\mathbf{e_j}||_2}$$

- Obviamente los conceptos de distancia y similaridad están relacionados: a mayor distancia menor similaridad.
- Existen diferentes formas para relacionar ambas medidas:
  - Distancia complemento: $d_{ij} = 1 - s_{ij}$
  - Raiz del complemento del cuadrado: $d_{ij} = \sqrt{1 - s_{ij}^2}$

Distancia y similaridad

- Generalmente cada elemento del conjunto Ω tendrá asociada una variable y podrá ser representado mediante el punto $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ en el espacio $\mathbb{R}^n$.
- Dependiendo de la naturaleza de las variables, se deberán utilizar diferentes tipos de distancias y similaridades.
- A continuación comentaremos las más habituales.

Distancia para variables continuas I

- Sean $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ e $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ dos elementos del conjunto Ω, en el que todas las variables son continuas, las medidas de distancia más utilizadas son:

Función de Distancia	**Fórmula**
Euclidea	$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
Manhattan	$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Norma del supremo	$d(\mathbf{x}, \mathbf{y}) = \sup_{i \in \{1,2,\ldots,n\}} \lvert x_i - y_i \rvert$
Minkosky	$d(\mathbf{x}, \mathbf{y}) = \sqrt[p]{\sum_{i=1}^{n}(x_i - y_i)^p}\ p > 0$
Mahanalobis	$d(\mathbf{x}, \mathbf{y}) = \sqrt{[(\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})]}$ Σ Matriz de covarianzas

- Para evitar que unas variables dominen sobre otras, las variables continuas se suelen normalizar.

Distancia para variables continuas II

- Para la normalización se suele utilizar el z-score.
- Sea x_{ij} el valor de la variable j en la instancia i, el valor normalizado z_{ij} se calcula de la siguiente forma:

$$z_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

con

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \quad \text{y} \quad \sigma_j = \sqrt{\frac{\sum_{i=1}^{n}(x_{ij} - \mu_j)^2}{n-1}}$$

Similaridad para variables binarias

- Sean $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ e $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$ dos elementos del conjunto Ω, en el que todas las variables son binarias
- En este caso es más fácil calcular primero la similitud, para después transformarla en distancia.
- Para ello se calcula la matriz de confusión para calcular las coincidencias entre las m variables:

		x_i		
		1	0	
y_i	1	a	b	a+b
	0	c	d	c+d
		a+c	b+d	m

Similaridad para variables binarias

Función de Distancia	**Fórmula**
Índice de Acuerdo	$s(\mathbf{x}, \mathbf{y}) = \frac{a+d}{m}$
Jaccard	$s(\mathbf{x}, \mathbf{y}) = \frac{a}{a+b+c}$
Russel-Roo	$s(\mathbf{x}, \mathbf{y}) = \frac{a}{m}$
Czekanowski	$s(\mathbf{x}, \mathbf{y}) = \frac{2a}{2a+b+c}$

Otro tipo de variables

- Para el caso de variables cualitativas o nominales existen dos posibilidades:
  - Contando las coincidencias:

$$d(\mathbf{x}, \mathbf{y}) = \frac{m - p}{m}$$

    siendo m el número total de variables y p el número de coincidencias.
  - Creando un atributo binario para cada uno de los posibles valores y calcular la similaridad como se describió anteriormente.

Otro tipo de variables

- Para el caso de variables ordinales (en este caso el orden si es importante) estas se tratan como numéricas después calcular su correspondencia al intervalo $[0, 1]$:

$$z_{ij} = \frac{x'_{ij} - 1}{M_k - 1}$$

siendo :

  - z_{ij} el valor transformado para el objeto i de la variable j,
  - x'_{ij} el valor entero, mayor que 1, que indica el orden que ocupa el valor x_{ij} entre los valores ordinales en el dominio de la variable j.
  - M_j el límite superior del dominio de la variable j (se asume que el límite inferior es 1).

Similaridad para variables mixtas

- Cuando se tiene variables de diferentes tipos se puede utilizar cualquier medida de agregación de las distancias/similaridades de las variables independientes.

$$d(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{n} \omega_l d(x_l, y_l) \text{ con } \sum_{l=1}^{n} \omega_l = 1$$

- donde:
  - $d(x_l, y_l)$ se corresponden con las distancias de cada una de las variables.
  - ω_l es el peso asociado a cada una de las variables y se tiene que cumplir .

Agrupamiento Jerárquico

- Un agrupamiento jerárquico es una sucesión de particiones "anidadas":
  - Cada grupo de elementos pertenecientes a una partición está totalmente incluido en alguna partición de nivel superior.
  - Esta estructura tiene una representación gráfica muy intuitiva denominada "dendograma".
  - El dendograma representa cómo se van agrupando los elementos en diferentes grupos (clusters) de forma anidada.
  - Se representa mediante un árbol binario en el que los elementos individuales se encuentran en los nodos hojas y los nodos intermedios representan diferentes agrupaciones de elementos.

Agrupamiento Jerárquico

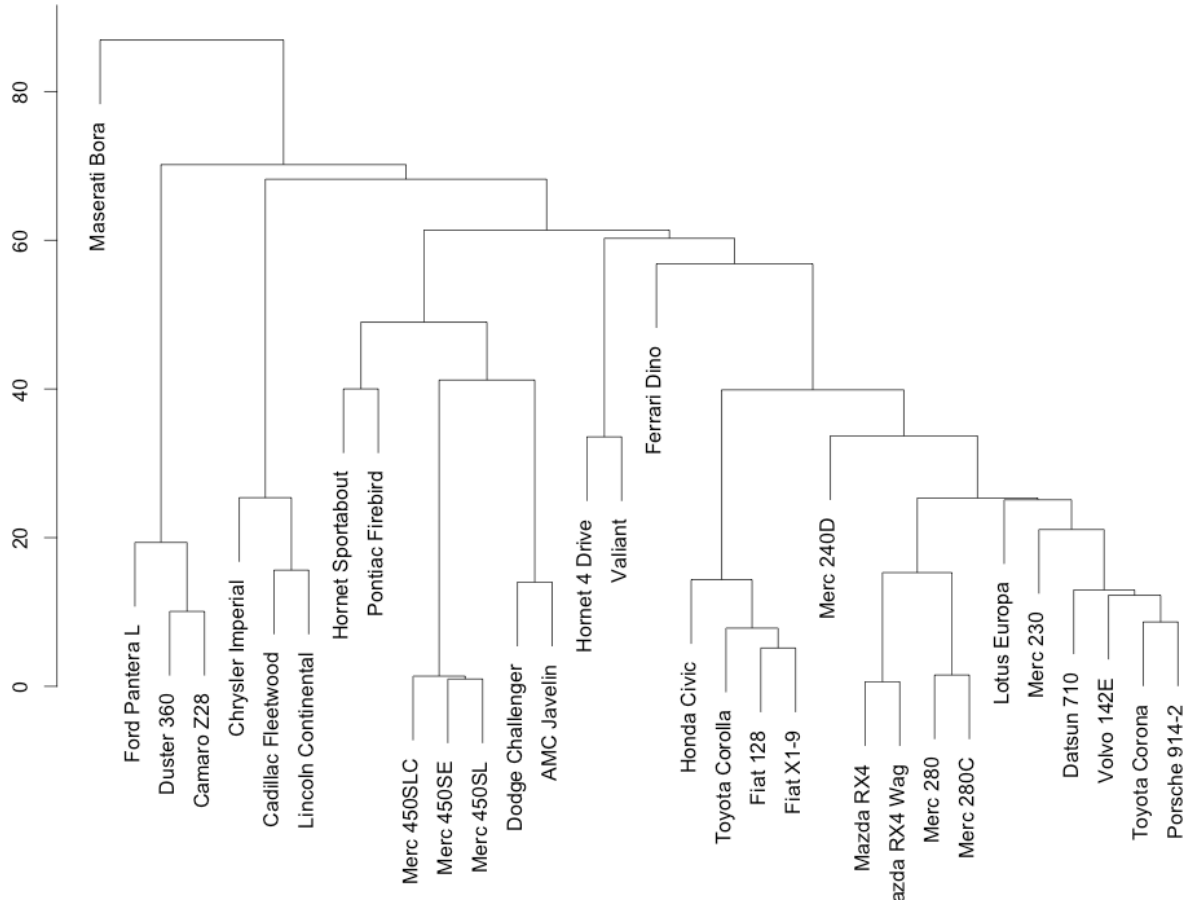


Figura: Dendograma

Agrupamiento Jerárquico

- Existen dos tipos de técnicas para construir agrupamientos jerárquicos:
  - Las técnicas **aglomerativas** generan nuevos clusters uniendo clusters similares.
    - Se parte de una partición inicial en la que cada elemento forma un cluster.
    - Se van uniendo de dos en dos aquellos clusters que están más próximos.
    - Finaliza el proceso cuando todos los elementos están ubicados en un único cluster
  - Las técnicas **divisivas** los nuevos clusters se generan dividiendo clusters.
    - Se parte de un único cluster que contiene todos los elementos.
    - Se va dividiendo dicho cluster hasta alcanzar una partición en la que todos los clusters contiene un único elemento.

Agrupamiento Jerárquico

- Las técnicas aglomerativas son más eficientes que las divisivas.
- Las técnicas divisivas tienen la ventaja de que parten de la información global que hay en los datos y no tienen por qué llegar hasta clusters de tamaño 1.
- Sin embargo, las técnicas divisivas son muy lentas y, sólo se utilizan en el caso de que existan pocos datos.
- Esto hace que los métodos más utilizados sean los aglomerativos, que son los que vamos a analizar a continuación.

Agrupamiento Jerárquico: Ejemplo

	E_1	E_2	E_3	E_4	E_5	E_6	E_7
E_1	0	10	10	7	6	13	8
E_2		0	4	7	8	**2**	8
E_3			0	9	12	3	8
E_4				0	5	5	5
E_5					0	6	6
E_6						0	9
E_7							0

Agrupamiento Jerárquico: Ejemplo

	E_1	E_3	E_4	E_5	E_7	(E_2, E_6)
E_1	0	10	7	6	8	10
E_3		0	9	12	8	**3**
E_4			0	5	5	5
E_5				0	6	6
E_7					0	8
(E_2, E_6)						0

Agrupamiento Jerárquico: Ejemplo

	E_1	E_4	E_5	E_7	$((E_2, E_6), E_3)$
E_1	0	7	6	8	10
E_4		0	**5**	5	5
E_5			0	6	6
E_7				0	8
$((E_2, E_6), E_3)$					0

Agrupamiento Jerárquico: Ejemplo

	E_1	(E_4, E_5)	E_7	$((E_2, E_6), E_3)$
E_1	0	6	8	10
(E_4, E_5)		0	**5**	5
E_7			0	8
$((E_2, E_6), E_3)$				0

Agrupamiento Jerárquico: Ejemplo

	E_1	$((E_4, E_5), E_7)$	$((E_2, E_6), E_3)$
E_1	0	6	10
$((E_4, E_5), E_7)$		0	**5**
$((E_2, E_6), E_3)$			0

Agrupamiento Jerárquico: Ejemplo

	E_1	$((E_4, E_5), E_7), ((E_2, E_6), E_3))$
E_1	0	6
$((E_4, E_5), E_7), ((E_2, E_6), E_3))$		0

Agrupamiento Jerárquico: Ejemplo

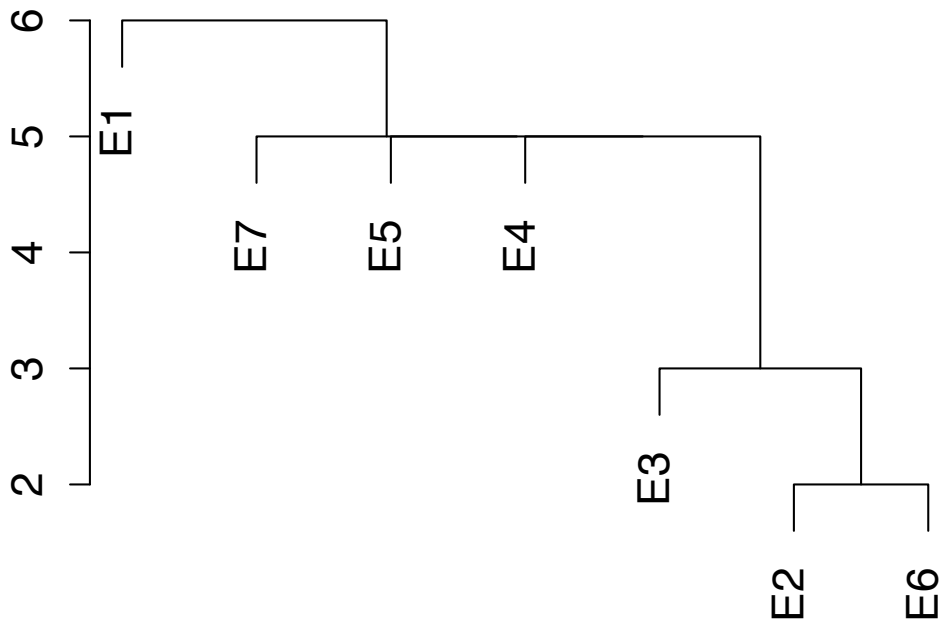


Figura: Dendograma resultante del ejemplo.

Agrupamiento Jerárquico: Distancia entre clusters I

- Como se puede ver, la clave de un agrupamiento jerárquico reside en la forma en que se defina la distancia entre dos clusters.
- Sean A y B dos clusters, la distancia entre ambos,$d(A, B)$ se puede definir de diferentes formas:
  - **Método de enlace simple** ("single link"). En este caso la distancia $d(A, B)$ se calcula como la distancia mínima entre los elementos de ambos clusters:

$$d(A, B) = \min_{\mathbf{x} \in A, \mathbf{y} \in B} d(\mathbf{x}, \mathbf{y})$$

  - **Método del enlace completo** ("complete link"). En este caso la distancia $d(A, B)$ se calcula como la distancia máxima entre los elementos de ambos clusters:

$$d(A, B) = \max_{\mathbf{x} \in A, \mathbf{y} \in B} d(\mathbf{x}, \mathbf{y})$$

Agrupamiento Jerárquico: Distancia entre clusters II

- **Método del enlace promedio** ("average link"). En este caso la distancia $d(A, B)$ se calcula como el promedio de la distancia entre cada par de elementos de ambos clusters:

$$d(A, B) = \frac{1}{|A||B|} \sum_{\mathbf{x} \in A, \mathbf{y} \in B} d(\mathbf{x}, \mathbf{y})$$

- **El método del centroide**. En este caso la distancia $d(A, B)$ se calcula como la distancia entre los centroides de cada grupo. El centroide del cluster A se calcula de la siguiente forma

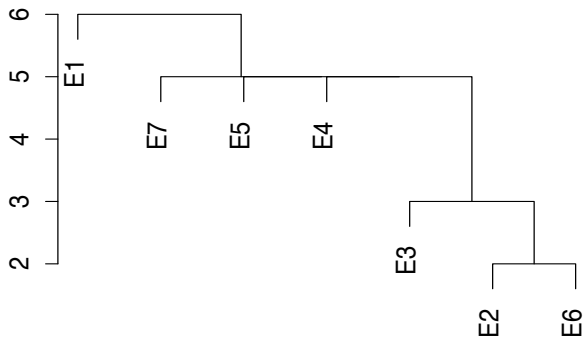
$$\overline{\mathbf{x}} = \frac{1}{|A|} \sum_{\mathbf{x} \in A} \mathbf{x}$$

- **El método del Ward**. En este caso se trata de fusionar aquellos clusters de tal forma que en el nuevo cluster la suma de las distancias de los elementos al centroide sea menor.

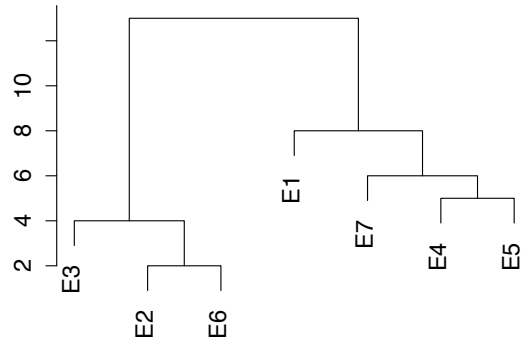
Agrupamiento Jerárquico: Distancia entre clusters III

- Existen versiones ponderadas de los métodos **promedio** y **centroide** que intentan compensar el hecho de fusionar cluster de tamaños muy dispares. Estos métodos se deberían utilizar cuando se sospeche de que el tamaño de los distintos clusters va a ser muy dispares.

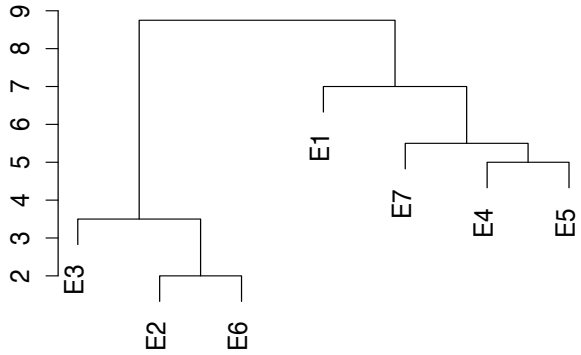
Agrupamiento Jerárquico: Ejemplo



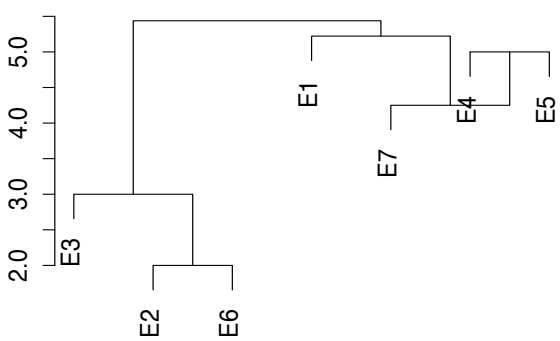
Enlace simple



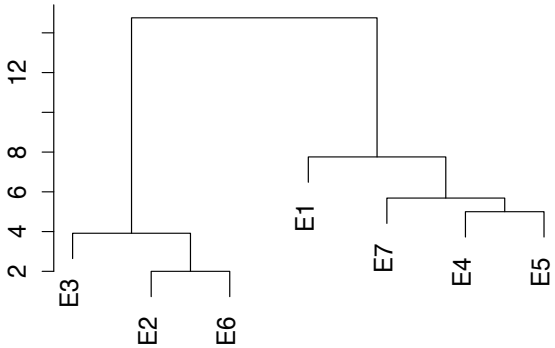
Enlace completo



Enlace Promedio



Centroides



Ward

Agrupamiento particional

- El objetivo de un agrupamiento particional es, dado un conjunto n elementos representados en un espacio $d-$dimensional:
  - encontrar una partición del mismo en k subconjuntos.
  - los elementos dentro de un grupo se tiene que parecer más entre sí que a los elementos de otros grupos
- El número k de subgrupos puede ser conocido apriori o no,
  - En la mayoría de las técnicas ese dato es un parámetro.

Agrupamiento particional

- Es necesario un criterio para medir la coherencia de cada grupo, así como la de entre grupos.
- Existen dos tipos de criterios:
  - Los métodos basados en **criterios globales** representan cada grupo mediante un prototipo, asignando cada elemento al grupo del prototipo más cercano. **K-medias** y **K-medoides** son técnicas que se corresponden con este tipo de criterio.
  - Los métodos basados en **criterios locales** forman grupos utilizando la estructura local de los datos, por ejemplo, identificando regiones de alta densidad de puntos. Uno de los métodos más conocidos dentro de este enfoque es **DBSCAN**.

Agrupamiento en base a enconders

- Supongamos que tenemos un conjunto de n elementos $X = \{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_n}\}$.
- El resultado de aplicar una técnica de agrupamiento para encontrar k clusters, $\{C_1, C_2, \ldots, C_k\}$ $(k < n)$, se puede definir mediante un "enconder" C:

Encoder

$$C(i) = t \Leftrightarrow \mathbf{x_i} \in C_t$$

es decir, el "encoder" C nos indica a qué cluster pertenece cada elemento.

- Por lo tanto, el objetivo de una técnica de agrupamiento debe ser encontrar el "encoder" $C^*(i)$ que optimice algún criterio determinado.

Distancia intra e intercluster I

- Sea C un encoder de k clusters sobre el conjunto $X = \{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_n}\}$.
- Hay que tener en cuenta que la separación total entre los elementos de X siempre es constante:

$$T = \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} d_{ii'} = \frac{1}{2}\sum_{j=1}^{k}\sum_{i:C(i)=j}\left(\sum_{i':C(i')=j} d_{ii'} + \sum_{i':C(i')\neq j} d_{ii'}\right)$$

con lo que en realidad tenemos $T = W(C) + B(C)$

Distancia intra e intercluster II

$$W(C) = \frac{1}{2}\sum_{j=1}^{k}\sum_{i:C(i)=j}\sum_{i':C(i')=j} d_{ii'}$$

$$B(C) = \frac{1}{2}\sum_{j=1}^{k}\sum_{i:C(i)=j}\sum_{i':C(i')\neq j} d_{ii'}$$

- Es decir, la distancia total en entre los puntos de un conjunto dividido en k clusters se puede calcular mediante la suma de la distancia entre los puntos de distintos clusters (intracluster, $W(C)$) y la distancia entre los puntos de cada cluster (intercluster, $B(C)$).

Distancia intra e intercluster III

- Por lo tanto, para obtener un buen agrupamiento (encoder) podemos:
  - Maximizar la distancia intercluster $B(C)$, buscar clusters lo más separados entre sí.
  - Minimizar la distancia intracluster $W(C)$, buscar clusters lo más compactos posibles.
- Ambas opciones son equivalentes ya que $W(C) = T - B(C)$

Método K-medias

- El algoritmo iterativo **K-medias** es el más popular que se puede aplicar a variables numéricas y la distancia euclídea:

$$d_{ij} = d(\mathbf{x_i}, \mathbf{x_j}) = \sum_{t=1}^{m} (x_{it} - x_{jt})^2 = ||\mathbf{x_i} - \mathbf{x_j}||^2$$

- Por tanto, la distancia intracluster puede escribirse como:

$$W(C) = \frac{1}{2} \sum_{l=1}^{k} \sum_{i:C(i)=l} \sum_{j:C(j)=l} ||\mathbf{x_i} - \mathbf{x_j}||^2$$

Método K-medias

Se puede demostrar que (ver Anexo I)

$$\frac{1}{2}\sum_{i:C(i)=l}\sum_{j:C(j)=l}||\mathbf{x_i}-\mathbf{x_j}||^2 = |C_l|\sum_{i:C(i)=l}||\mathbf{x_i}-\boldsymbol{\mu_l}||^2$$

Por lo tanto, minimizar la distancia intracluster $W(C)$ sería equivalente a minimizar la distancia a los centroides:

$$\sum_{l=1}^{k}\sum_{i:C(i)=l}||\mathbf{x_i}-\boldsymbol{\mu_l}||^2$$

es decir, la distancia de cada elemento a su respectivo centroide $\boldsymbol{\mu_l}$, donde

$$\boldsymbol{\mu_l} = \frac{1}{|C_l|}\sum_{i:C(i)=l}\mathbf{x_i}$$

Método K-medias

- De esta forma, podemos plantear el problema de optimización como:

$$C^* = \min_{C} \sum_{l=1}^{k} \sum_{i:C(i)=l} ||\mathbf{x_i} - \boldsymbol{\mu_l}||^2$$

- El algoritmo K-means es un algoritmo que intenta resolver este problema siguiendo un esquema de ascensión de colinas por la máxima pendiente.
  - Después de cada iteración no se puede volver atrás y probar otros centroides.

Algoritmo K-medias

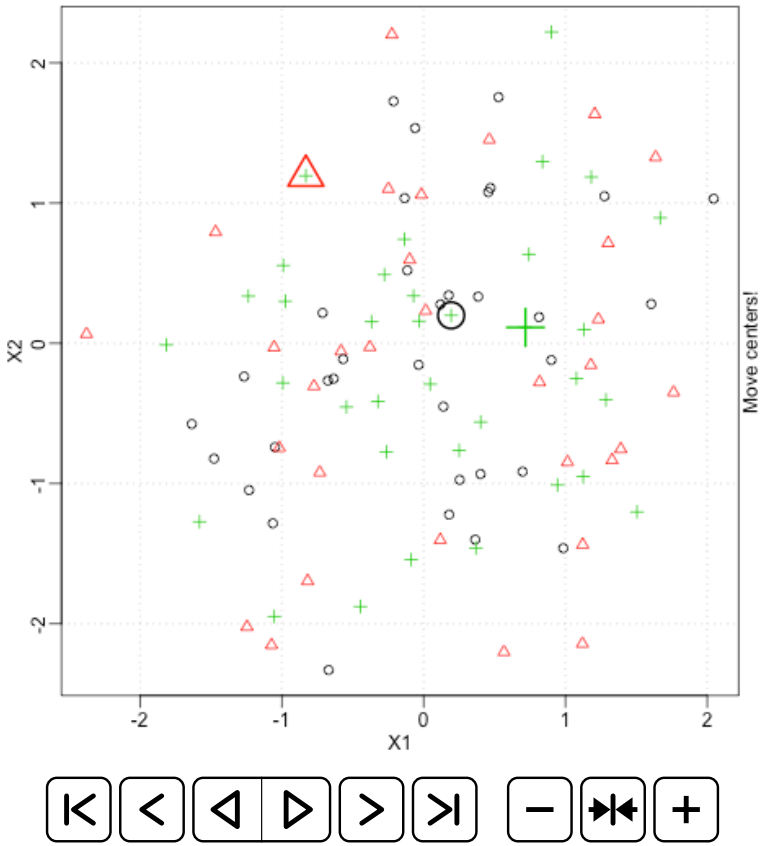
1. Comenzar con alguna de las dos configuraciones iniciales:
   - Si se inicializan aleatoriamente los centroides, $\boldsymbol{\mu_l}$ de cada uno de los k clusters, ir al paso 2.
   - Si se parte de una partición aleatoria del conjunto de entrada en k clusters, ir al paso 3.
2. Calcular el encoder (distribuir los elementos entre los k clusters de acuerdo a los los centroides, $\boldsymbol{\mu_l}$).

$$C(i) = arg \min_{1 \leq l \leq k} ||\mathbf{x_i} - \boldsymbol{\mu_l}||^2$$

3. Calcular los nuevos centroides $\boldsymbol{\mu_l}$ para $l = \{1, \ldots, k\}$.
4. Si los nuevos centroides no se han estabilizado, volver al paso 2, si no, fin

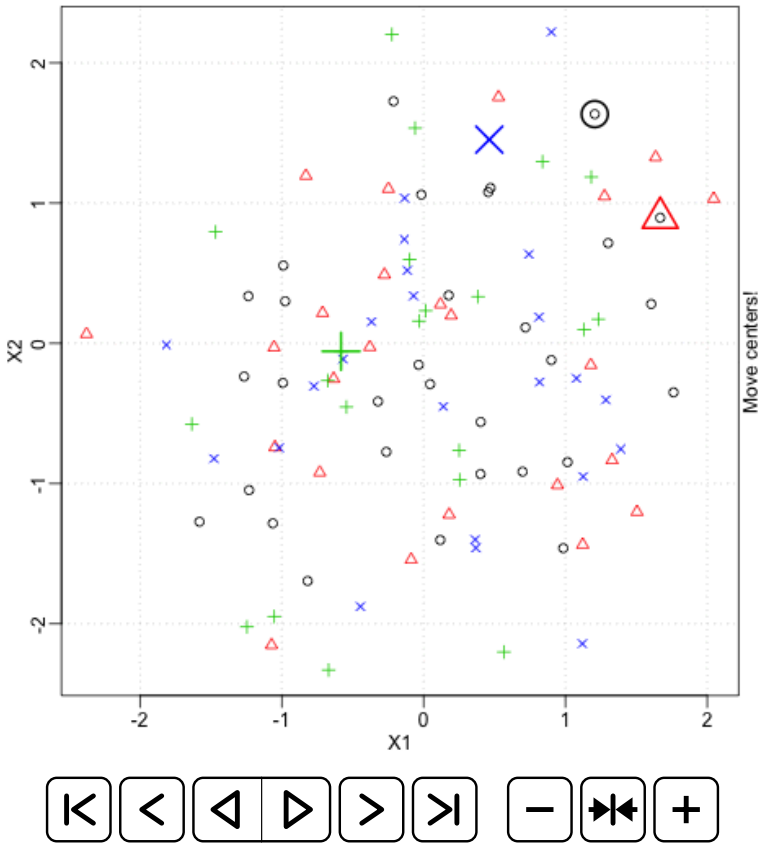
Método K-medias: Ejemplo

Ejemplo K-medias con 3 clusters



Método K-medias: Ejemplo

Ejemplo K-medias con 4 clusters



Método K-medias: Consideraciones

- El algoritmo K-medias se aplica en el caso de que todos los atributos sean reales.
- En principio hemos utilizado la distancia euclídea:
  - Los representantes se corresponden la media aritmética de los elementos del cluster.
  - Lo que se está minimizando el la desviación respecto a los representantes de cada cluster. Es decir, la distancia intracluster.
  - Sin embargo, amplifica el efecto de los outliers.
- Se pueden considerar otras medidas de distancia.
- También se pueden utilizar medidas de similitud con lo que en vez de minimzar habría que maximizar.

Método K-medias: Problemas I

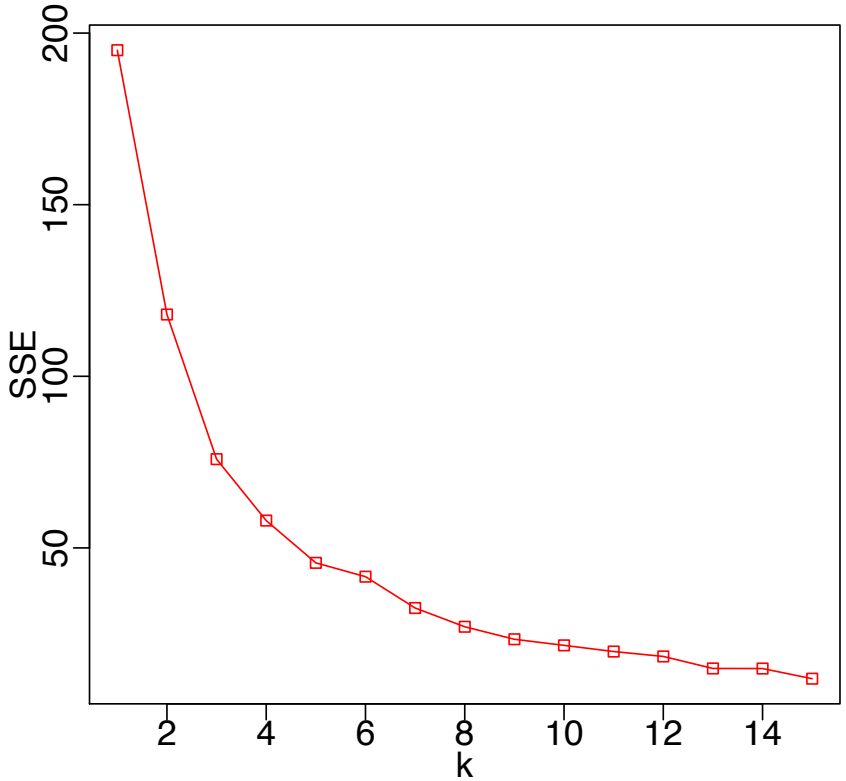
- Es muy sensible a la elección de los centroides
  - Se pueden realizar varias ejecuciones con diferentes centroides iniciales y comparar resultados.
  - El método más simple es escoger, de entre las n instancias del conjunto de observaciones, k instancias para inicializar los $\boldsymbol{\mu}_l$.
  - Una posibilidad algo más elaborada es usar el vector de medias $\boldsymbol{\mu}$ de todo el conjunto de datos. Para obtener cada centroide $\boldsymbol{\mu}_l$. sumamos o restamos valores aleatorios a cada una de sus componentes.
  - Hacer un análisis PCA (Principal Component Analysis), dividir el rango de la primera variable generada por PCA en k intervalos iguales y generar los centroides a partir de la media de dichos intervalos
  - Usar clustering jerárquico

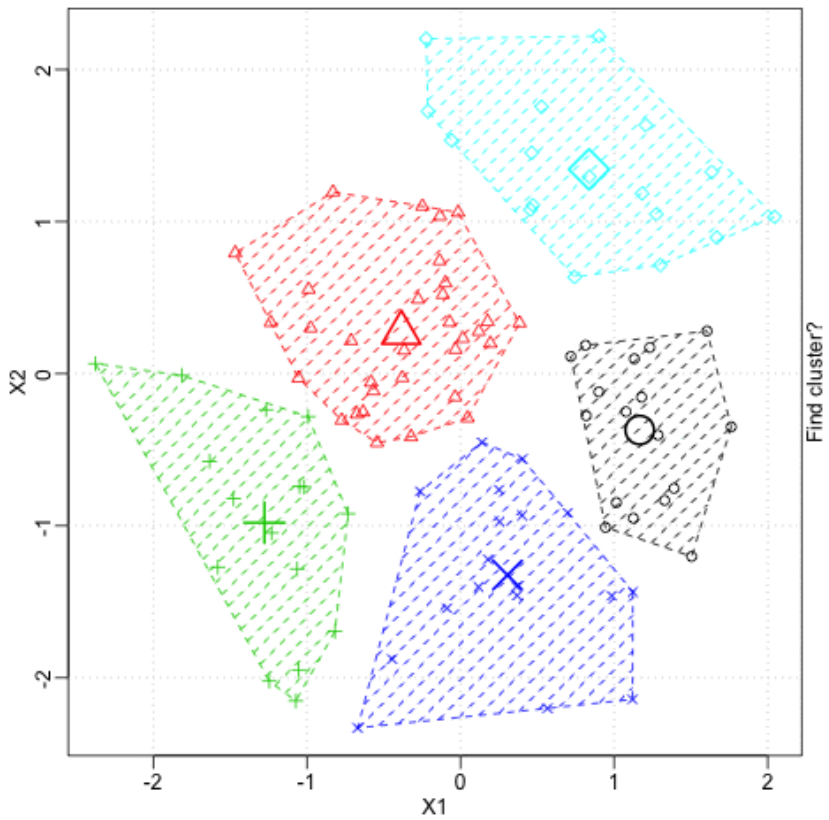
Método K-medias: Problemas II

- K-medias es muy sensible al número de clusters, k, y hay que elegirlos a priori.
  - Se puede usar un método jerárquico para estimar el valor de k.
  - Se pude aplicar K-medias para distintos valores de k y comprobar cuando no hay mejoras significativas del SSE.

Método K-medias: Problemas III

- En nuestro ejemplo, parece que el corte puede ser con $k = 5$





Método K-medias: Problemas IV

- Al usar la media para calcular los centroides el método es sensible a los outliers
  - Utilizar las medianas.
  - Eliminar los outliers (en algunos casos puenden ser de interés).
  - Utilizar K-medoides: el representante tiene que ser el elemento más representativo del cluster.
- Para manejar datos no numéricos se requiere la redifinición de la función de distancia, trabajar con la moda y no la media.
- K-medias no funciona bien cuando los clusters son de: distinto tamaño, diferente densidad y no convexos.
  - Esto requiere una revisión posterior de los resultados y hacer varias pruebas para distintos valores de K.

Método K-medoides

- Para evitar la sensibilidad del método K-medias a los outliers, K-medoides elige como representante de cada cluster a un punto del mismo considerado más representativo, la **mediana**.
- Al no basarse en los centroides (valores medios), no hace falta la definición de una función de distancia ya que puede operar directamente con la matriz de distancias (o similaridad).
- El proceso es idéntico al método K-medias sólo que el cálculo de los centroides se sustituye por el de los medoides.
- Es más costoso computacionalmente que el método K-medias, además de necesitar también saber a priori el número de grupos

Algoritmo K-medoides

1. Comenzar con alguna de las dos configuraciones iniciales:
   - Si escogen aleatoriamente k elementos como representantes m_i de los k clusters, ir al paso 2.
   - Si se parte de una partición aleatoria del conjunto de entrada en k grupos, ir al paso 3.
2. Calcular el encoder (distribuir los elementos entre los k clusters de acuerdo a los representantes m_l).

$$C(i) = arg \min_{1 \leq l \leq k} d(\mathbf{x_i}, \mathbf{m_l})$$

3. Calcular los nuevos mediodes, $\mathbf{m_l}$ con $l = 1, \ldots, k$, de cada cluster:

$$\mathbf{m_l} = \arg \min_{i:C(i)=l} \sum_{j:C(j)=l} d(\mathbf{x_i}, \mathbf{x_j})$$

4. Si los nuevos medoides, $\mathbf{m_l}$ no se han estabilizado, volver al paso 2, si no, fin

Método K-medoides

- Al utilizar las medianas en vez valores medios, se seleccionan como representantes elementos del conjunto de datos:
  - Esto permite que el método sea más **robusto** y no se ve tan afectado por la presencia de outliers
  - También se gana en **interpretabilidad**.
- El problema que plantea es el alto coste computacional.
  - Funciona bien para conjunto relativamente pequeños de datos y pocos clusters, comparado con K-medias,

Método K-medoides

Existen varias implementaciones:

- **PAM** (Partition Arround Medoids) consiste en la implementación de las ideas anteriormente propuestas.
- **CLARA** (Clustering LARge Applications) intenta reducir la carga computacional de PAM seleccionando los medoides de una muestra aleatoria y significativa de los datos y despues aplica PAM. Básicamente realiza varios muestreos y da como resultado el mejor clustering.
- **CLARANS** que se diferencia del anterior en que la búsqueda de los medoides se aproxima como un proceso de búsqueda, realizando un muestreo cada vez que se calcula un medoide.

Método DBSCAN

- El método **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) es un método basado en criterios locales que se apoya en el concepto de densidad de los puntos.
  - Se consideran clusters aquellas regiones del espacio con una alta densidad de puntos.
  - Las regiones con una baja densidad se podrán corresponder con puntos que no están asociados a la mayoría de los datos.

Método DBSCAN: Conceptos previos I

- Para describir el algoritmo necesitamos definir los siguientes conceptos:

$\epsilon-$vecindad

Sean $\mathbf{x_k}$ un elemento del conjunto de datos $\Omega = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ y $\epsilon > 0$ con $\epsilon \in \mathbb{R}$, la $\epsilon-$vecindad del punto $\mathbf{x_k}$, $N_\epsilon(\mathbf{x_k})$, se define como:

$$N_\epsilon(\mathbf{x_k}) = \{\mathbf{x} \in \Omega | d(\mathbf{x}, \mathbf{x_k}) \leq \epsilon\}$$

es decir, la $\epsilon-$vecindad de un punto incluye todos aquellos puntos que están a una distancia menor o igual que ϵ.

- La geometría de la la $\epsilon-$vecindad vendrá determinada por la medida de distancia utilizada.

Método DBSCAN: Conceptos previos II

- Para determinar cuando una $\epsilon-$vecindad tiene una densidad alta se define el parámetro *MinPts*, de tal forma que si:

$$|N_\epsilon(\mathbf{x_k})| \geq MinPts$$

diremos que la densidad entorno al punto $\mathbf{x_k}$ es alta y $\mathbf{x_k}$ es un **punto núcleo**.

- Todo punto dentro de la $\epsilon-$vecindad de un punto núcleo se denomina **punto frontera**. Un punto frontera puede pertenecer a más de una $\epsilon-$vecindades distintas.
- El resto de puntos se consideran **puntos ruido**.

Método DBSCAN: Conceptos previos III

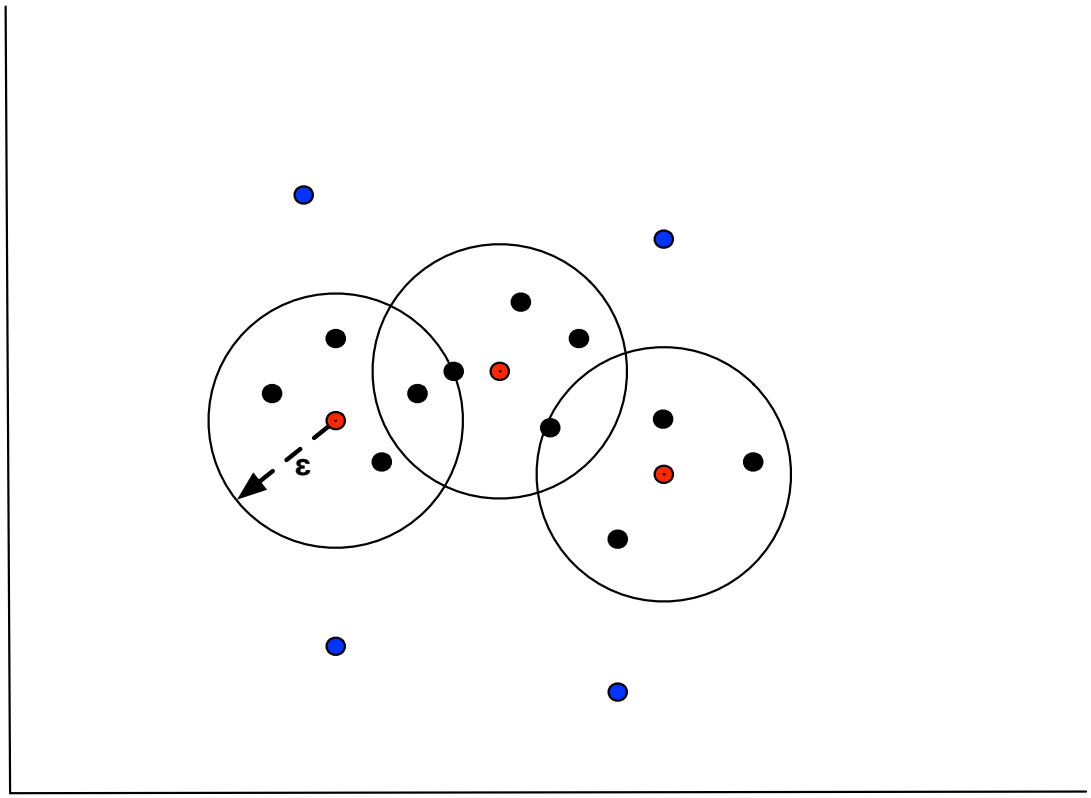


Figura: Concepto de $\epsilon-$ vecindad y puntos núcleo (rojo), puntos frontera (negro) y puntos ruido (azul) (MinPts $= 4$).

Método DBSCAN: Conceptos previos IV

Densidad alcanzable directa

Sean $\mathbf{p}$ y $\mathbf{q}$ dos elementos del conjunto Ω, decimos que $\mathbf{q}$ es directamente densidad alcanzable desde $\mathbf{p}$, si y sólo si:

1. $\mathbf{q} \in N_\epsilon(\mathbf{p})$
2. $\mathbf{p}$ es un punto núcleo.

Densidad alcanzable

Sean $\mathbf{p}$ y $\mathbf{q}$ dos elementos del conjunto Ω, decimos que $\mathbf{q}$ es densidad alcanzable desde $\mathbf{p}$, si existe una cadena de puntos $\mathbf{p_1}, \mathbf{p_2}, .., \mathbf{p_l}$ tal que:

1. $\mathbf{p_1} = \mathbf{p}$ y $\mathbf{p_l} = \mathbf{q}$
2. $\forall i \in \{1, \ldots, l\}$ $\mathbf{p_{i+1}}$ es directamente densidad alcanzable desde $\mathbf{p_i}$

Es transitiva pero no simétrica

Método DBSCAN: Conceptos previos V

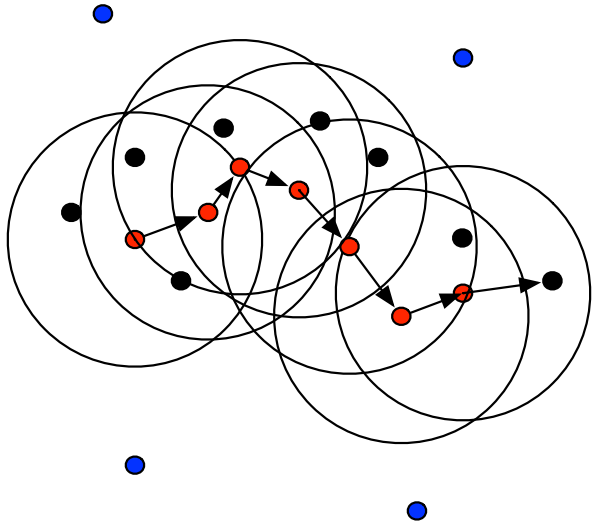


Figura: Concepto de densidad alcanzable

Método DBSCAN: Conceptos previos VI

Puntos densamente conectados

Sean **p** y **q** dos elementos del conjunto Ω, decimos que **p** y **q** están densamente conectados si son directamente alcanzables desde un mismo punto **o**.

La conectividad densa es simétrica.

Cluster

Sean un conjunto Ω y los parámetros ϵ y *MinPts*, un cluster C_l, es un subconjunto de Ω que satisface los siguientes criterios:

1. **Maximalidad:** $\forall \mathbf{p}, \mathbf{q}$, si $\mathbf{p} \in C_l$ y **q** es densidad alcanzable desde **p**, entonces $\mathbf{q} \in C_l$
2. **Conectividad:** $\forall \mathbf{p}, \mathbf{q} \in C_l$, **p** y **q** están densamente conectados

Un cluster contiene puntos núcleo y puntos frontera.

Algoritmo DBSCAN I

- La idea básica del método DBSCAN consiste en crear clusters con todos los puntos que son densidad alcanzable.
  1. Se especifican los parámetros ϵ y *MinPts*.
  2. Seleccionar arbitrariamente un punto, $\mathbf{x_k} \in \Omega$.
  3. Encontrar todos aquellos puntos densidad alcanzables desde $\mathbf{x_k}$.
  4. Si x_k es un punto núcleo, se forma un cluster y se incluyen también los puntos en su $\epsilon-$vecindad.
     - Se intenta expandir añadiendo todos los puntos densidad alcanzable a otros puntos núcleos del cluster
  5. Si $\mathbf{x_k}$ es un punto frontera se procede con el siguiente punto.
  6. En otro caso, el punto se etiqueta como ruido y se desecha.

Algoritmo DBSCAN II

- Los pasos 2-5 se repiten hasta que todos los puntos han sido visitados o añadidos a algún cluster.
- Básicamente se añaden al cluster todos aquellos puntos densamente conectados desde los puntos del cluster. Esto permite una gran cantidad de geometrías diferentes para los clusters.
- Sin embargo, hay tres parámetros que influyen notablemente en el método:
  - La función de distancia elegida, que definirá la geometría de la $\epsilon-$vecindad.
  - Valores altos para ϵ requieren valores altos para *MinPts*.
  - Un valor bajo para ϵ dará lugar a un número alto de clusters pequeños. A medida que se vaya aumentando dicho valor se irán produciendo un número más pequeño de clusters, pero aumentará el número de puntos ruido.

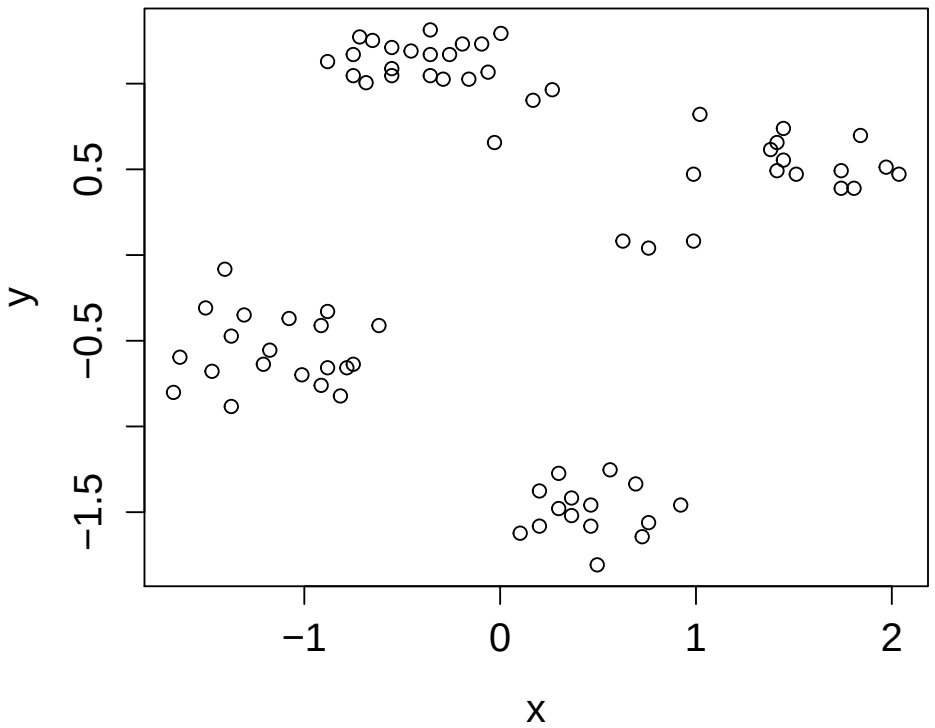
Método DBSCAN: Selección de parámetros I

- El parámetro *MinPts* se suele fijar a $MinPts = d + 1$, siendo d el número de dimensiones (algunos autores) utilizan $MinPts = 2d - 1$.
  - El valor $MinPts = 1$ no tiene sentido.
  - El valor $MinPts = 2$ equivale a un agrupamiento jerárquico de enlace simple cortado a la altura ϵ.
  - Los valores grandes de *MinPts* son generalmente mejores para datos con ruido.
  - A medida que el conjunto de datos sea mayor *MinPts* debe ser mayor.

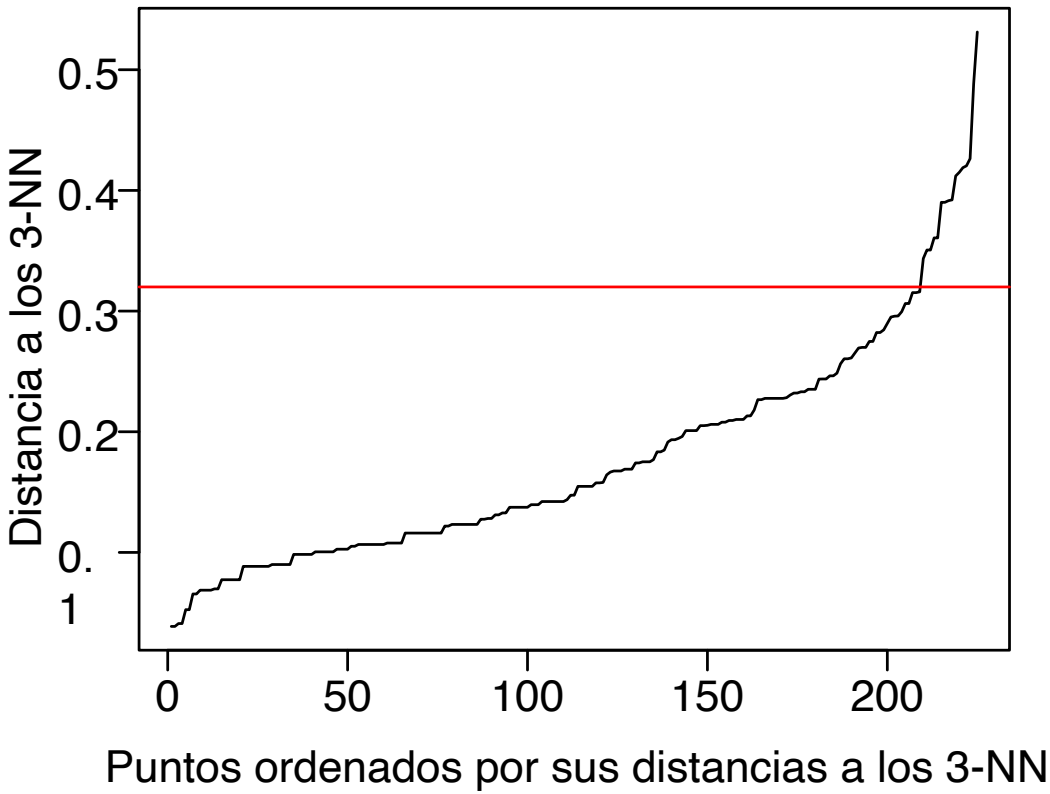
Método DBSCAN: Selección de parámetros II

- El valor ϵ puede ser elegido por medio de un gráfico de k-distancias con $k = MinPts$.
  - Se fija el valor de ϵ a la distancias en la que se muestre una fuerte curvatura (es decir, un codo).
  - A medida que ϵ se va haciendo más grande, el tamaño de los clusters obtenidos aumentará.

Método DBSCAN: Ejemplo I

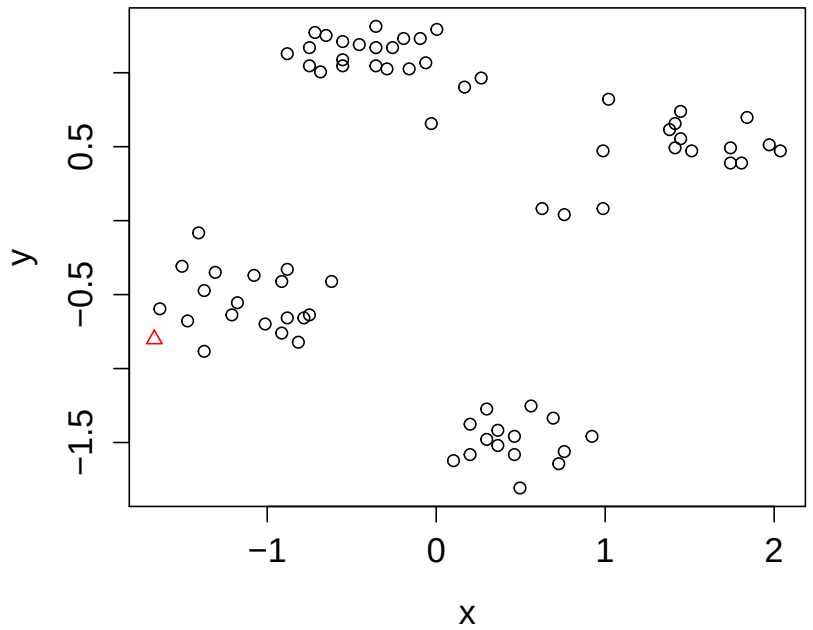


Método DBSCAN: Ejemplo II



Método DBSCAN: Ejemplo III

Ejemplo DBSCAN con $\epsilon = 0{,}32$ y $MinPts = 3$.



Método DBSCAN: Conclusiones

- **Ventajas:**
  - Los clusters pueden tener formas y tamaños arbitrarios.
  - El número de clusters se determina automáticamente.
  - Puede detectar y aislar el ruido, siendo robusto a los outlyers.
  - Se puede optimizar utilizando estructuras de datos para los índices (por ejemplo árboles K-D).
- **Desventajas:**
  - No es enteramente determinista, un punto frontera puede pertenecer a dos clusters distintos.
  - Los parámetros necesarios pueden ser difíciles de encontrar.
  - Es muy sensible a los valores de dichos parámetros.
- OPTICS (Ordering points to identify the clustering structure) es una generalización de DBSCAN en la que sólo se fija el parámetro *MinPts*

Método OPTICS: Generalidades

- OPTICS sólo requiere el parámetro *MinPts*.
- No genera un conjunto de clusters
  - Ordena los elementos del conjunto de datos de tal forma que aquellos puntos cercanos son vecinos en dicha ordenación.
  - También se almacena la distancia que se necesita para que dichos puntos pertenezcan al mismo cluster.
- La información sobre la ordenación y la distancia es equivalente a un DBSCAN para distintos valores de ϵ.
- Por lo tanto, se puede utilizar tanto de forma automática como interactiva a la hora de encontrar un clustering en el conjunto de datos.

Evaluación de los agrupamientos I

- Una vez aplicada una técnica de agrupamiento concreta:
  - ¿Son los clusters generados un fiel reflejo de la verdadera naturaleza de los datos?.
- Por regla general, la mayoría de las técnicas se ven influenciadas por dos parámetros:
  - La medida de distancia utilizada.
  - El número de clusters que la técnica concreta debe buscar.
- Esto nos lleva a que generalmente deberíamos elegir entre varias configuraciones posibles después de probar varias combinaciones de parámetros.
- Esta tarea se puede llevar a cabo por medio de una **medida de la calidad del agrupamiento**.
- Una medida de calidad trata de evaluar cómo se de buena es la estructura revelada por el agrupamiento obtenido respecto a la estructura real que presentan los datos.

Evaluación de los agrupamientos II

- De todas formas, hay que tener en cuenta que el éxito de la medida de calidad seleccionada depende de la técnica utilizada y la propias características de los datos.
- Atendiendo al resultado de una técnica de agrupamiento, este debe satisfacer dos propiedades:
  - **Compactación:** nos indica cuán cerca están entre sí los elementos de un cluster. A mayor varianza entre los elementos de un cluster menos compacto será este y, al contrario, a menor varianza mas compacto será.
  - **Separabilidad:** nos indica cuán distintos son los clusters entre sí.
- Una forma intuitiva de medir estas características puede ser las distancias intra e intercluster.
  - Un agrupamiento compacto y separable se debe caracterizar por una distancia intracluster pequeña y una distancia intercluster grande.

Índice Silueta (Silhouette index) I

- Supongamos que la observación $\mathbf{x_i}$ pertenece al cluster C_l, se define su índice silueta, $s(\mathbf{x_i})$, de la siguiente forma:

$$s(\mathbf{x_i}) = \frac{b(\mathbf{x_i}) - a(\mathbf{x_i})}{max(a(\mathbf{x_i}), b(\mathbf{x_i}))}$$

donde:

- $a(\mathbf{x_i})$ la distancia media entre $\mathbf{x_i}$ y todos los elementos de su mismo cluster, C_l.

$$a(\mathbf{x_i}) = \frac{1}{|C_l|} \sum_{\forall \mathbf{x_j} \in C_l \wedge j \neq i} d(\mathbf{x_i}, \mathbf{x_j})$$

Índice Silueta (Silhouette index) II

- Si el cluster C_l tiene un sólo elemento entonces $s(\mathbf{x_i}) = 0$
- $b(\mathbf{x_i})$ la distancia media entre $\mathbf{x_i}$ y todos los elementos del cluster más cercano,

$$b(\mathbf{x_i}) = min_{k \neq l}(d(\mathbf{x_i}, C_k)) \text{ con } d(\mathbf{x_i}, C_k) = \frac{1}{|C_k|} \sum_{\forall \mathbf{x_j} \in C_k} d(\mathbf{x_i}, \mathbf{x_j})$$

Índice Silueta (Silhouette index) III

- El índice silueta varía entre $[-1, 1]$.
- Un $s(\mathbf{x_i})$ cercano a 1 indica que el observación está muy bien agrupado.
- Un $s(\mathbf{x_i})$ cercano a 0 indica que el observación está entre dos clusters.
- Un $s(\mathbf{x_i})$ negativo indica que la observación está mal agrupada.
- El coeficiente silueta para todo el agrupamiento sería la media de todos los índices siluetas

Índice Gap I

- El índice gap [Tibshirani *et al.*, 2001] compara la varianza total intra-cluster observada para diferentes valores de k con el valor esperado en una distribución uniforme de referencia.
- Supongamos que nuestros datos han sido agrupados en k clusters $\{C_1, C_2, \cdots, C_k\}$, con $n_r = |C_r|$.

Índice Gap II

- Sea D_r la suma de la distancia entre todos los elementos del cluster r:

$$D_r = \sum_{\mathbf{x_i}, \mathbf{x_j} \in C_r} d(\mathbf{x_i}, \mathbf{x_j})$$

- Sea W_k la distancia intra-cluster para k número de clusters

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r$$

Índice Gap III

- **Algoritmo:**
  1. Calcular W_k para distintos valores de k
  2. Generar B conjuntos de referencia usando un muestreo uniforme
  3. Calcular la suma de la distancia intra-cluster, W^*_{bk} en cada uno de los B conjuntos y para distintos valores de k.
  4. El índice Gap se calcula como

$$Gap(k) = \frac{1}{B}\sum_b log(W^*_{kb}) - log(W_k)$$

  5. Sea $\bar{l} = \frac{1}{B}\sum_b log(W^*_{kb})$, calcular las desviaciones estandares s_k como:

$$s_k = \sqrt{\frac{1}{B}\sum_b (log(W^*_{kb}) - \bar{l})^2}$$

Índice Gap IV

6. Se determina el valor óptimo de k como:

$$k_{opt} = \min_{k}(gap(k) \geq gap(k+1) - s_k)$$

- Existen otros criterios pero este criterio ha demostrado experimentalmente mejor comportamiento.

Índice Davies-Bouldin I

- Sea un agrupamiento de k clusters $\{C_1, C_2, \ldots, C_k\}$.
- La distancia intracluster para cada cluster se puede definir como:

$$w_i = \frac{1}{|C_l|} \sum_{\mathbf{x_i} \in C_l} d(\mathbf{x_i}, \boldsymbol{\mu_l})$$

- donde $\{\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu_k}\}$ son los centroides de cada cluster.
- La distancia entre centroides se define como:

$$d_{ij} = d(\boldsymbol{\mu_i}, \boldsymbol{\mu_1})$$

- Si para cada cluster, calculamos el siguiente ratio;

$$r_i = \max_{j, j \neq i} \frac{w_i + w_j}{d_{ij}}$$

Índice Davies-Bouldin II

- El índice de Davies-Bouldin es la media de los ratios, r_i:

$$r = \frac{1}{c} \sum_{i=1}^{c} r_i$$

- Según este índice el valor óptimo para el número de clusters es aquel que hace mínimo el índice.
- Hay que tener en cuenta el valor de r mínimo se consigue con valores pequeños el numerador de r_i y valores grandes en el denominador.
  - Es decir, favorece la creación de agrupamientos compactos y separados.

Resumen I

- En este capítulo hemos abordado las técnicas principales para aprendizaje no supervisado, centrándonos en las técnicas de agrupamiento o clustering.
- Se ha presentado el concepto de distancia y similaridad como elemento clave para definir los agrupamientos, así como diferentes formas de medirlos
- Primero hemos analizado las técnicas de agrupamiento jerárquico y sus distintas variantes dependiendo de cómo se calcule la distancia entre grupos.

Resumen II

- El segundo grupo de técnicas que se han analizado corresponde con las técnicas particionales, que se han dividido en dos grupos:
  - Técnicas basadas en criterios globlales: K-medias y K-medoides.
  - Técnicas basadas en criterios locales como DBSCAN
- Por último, se han analizado algunas medidas para medir la calidad del agrupamiento obtenido.

Referencias I

Krzysztof J Cios, Witold Pedrycz, and Roman W Swiniarski.

Data mining methods for knowledge discovery, volume 458.

Springer Science & Business Media, 2012.

Roque Luis Marín Morales and José Tomás Palma Méndez, editors.

Inteligencia artificial: técnicas, métodos y aplicaciones.

McGraw-Hill, 2008.

Basilio Sierra Araujo.

Aprendizaje automático: conceptos básicos y avanzados: aspectos prácticos utilizando el software Weka.

Pearson Prentice Hall Madrid, 2006.

Robert Tibshirani, Guenther Walther, and Trevor Hastie.

Estimating the number of clusters in a dataset via the gap statistic.

Journal of the Royal Statistical Society B, 63:411–423, 2001.

Demostración reformulación K-medias

Si tenemos en cuenta que:

$$\sum_{i:C(i)=l}\sum_{j:C(j)=l}||\mathbf{x_i}-\mathbf{x_j}||^2=\sum_{i:C(i)=l}\sum_{j:C(j)=l}\left(||\mathbf{x_i}||^2+||\mathbf{x_j}||^2+2\langle\mathbf{x_i},\mathbf{x_j}\rangle\right)$$

$$=\sum_{i:C(i)=l}\left(\sum_{j:C(j)=l}||\mathbf{x_i}||^2+\sum_{j:C(j)=l}||\mathbf{x_j}||^2+2\sum_{j:C(j)=l}\langle\mathbf{x_i},\mathbf{x_j}\rangle\right) \quad (1)$$

y que:

$$\sum_{j:C(j)=l}\langle\mathbf{x_i},\mathbf{x_j}\rangle=\sum_{j:C(j)=l}\sum_{t=1}^{m}x_{it}\cdot x_{jt}=\sum_{t=1}^{m}x_{it}\cdot\sum_{j:C(j)=l}x_{jt}$$

$$=\sum_{t=1}^{m}x_{it}\cdot|C_l|\mu_{lt}=|C_l|\sum_{t=1}^{m}x_{it}\cdot\mu_{lt}=|C_l|\langle\mathbf{x_i},\boldsymbol{\mu_l}\rangle$$

Demostración reformulación K-medias

Sustituyendo en 1 tenemos

$$\sum_{i:C(i)=l}\sum_{j:C(j)=l}||\mathbf{x_i}-\mathbf{x_j}||^2=\sum_{i:C(i)=l}\left(\sum_{j:C(j)=l}||x_i||^2+\sum_{j:C(j)=l}||x_j||^2-2|C_l|\langle\mathbf{x_i},\boldsymbol{\mu_l}\rangle\right) \quad (2)$$
$$=2|C_l|\sum_{i:C(i)=l}||x_i||^2-2|C_l|\sum_{i:C(i)=l}\langle\mathbf{x_i},\boldsymbol{\mu_l}\rangle$$

como

$$\sum_{i:C(i)=l}\langle\mathbf{x_i},\boldsymbol{\mu_l}\rangle=\sum_{i:C(i)=l}\sum_{t=1}^{m}x_{it}\cdot\mu_{lt}=\sum_{t=1}^{m}\sum_{i:C(i)=l}x_{it}\cdot\mu_{lt}$$
$$=\sum_{t=1}^{m}|C_l|\mu_{lt}\cdot\mu_{lt}=|C_l|\sum_{t=1}^{m}\mu_{ip}^2=|C_l|||\boldsymbol{\mu_l}||^2$$

Demostración reformulación K-medias

Sustituyendo en 2 tenemos

$$\sum_{i:C(i)=l} \sum_{j:C(j)=l} ||\mathbf{x_i} - \mathbf{x_j}||^2 = 2|C_l| \sum_{i:C(i)=l} ||x_i||^2 - 2|C_l|^2 ||\boldsymbol{\mu_l}||^2 \tag{3}$$

dado que:

$$\begin{aligned} \sum_{i:C(i)=l} ||\mathbf{x_i} - \boldsymbol{\mu_l}||^2 &= \sum_{i:C(i)=l} ||x_i||^2 + \sum_{i:C(i)=l} ||\boldsymbol{\mu_l}||^2 - 2 \sum_{i:C(i)=l} \langle \mathbf{x_i}, \boldsymbol{\mu_l} \rangle \\ &= \sum_{i:C(i)=l} ||x_i||^2 + |C_l| ||\boldsymbol{\mu_l}||^2 - 2|C_l| ||\boldsymbol{\mu_l}||^2 \\ &= \sum_{i:C(i)=l} ||x_i||^2 + |C_l| ||\boldsymbol{\mu_l}||^2 \end{aligned} \tag{4}$$

Demostración reformulación K-medias

Igualando 2 y 4 tenemos

$$\frac{1}{2}\sum_{i:C(i)=l}\sum_{j:C(j)=l}||\mathbf{x_i}-\mathbf{x_j}||^2 = |C_l|\sum_{i:C(i)=l}||\mathbf{x_i}-\boldsymbol{\mu_l}||^2$$

Por lo tanto, minimizar la distancia intracluster $W(C)$ sería equivalente a minimizar

$$\sum_{l=1}^{k}\sum_{i:C(i)=l}||\mathbf{x_i}-\boldsymbol{\mu_l}||^2$$

es decir, la distancia de cada elemento a su respectivo centroide $\boldsymbol{\mu_l}$, donde

$$\boldsymbol{\mu_l} = \frac{1}{|C_l|}\sum_{i:C(i)=l}\mathbf{x_i}$$