

# Agrupamiento/Clustering

Tratamiento Inteligente de Datos  
Master Universitario en Ingeniería Informática



UNIVERSIDAD  
DE GRANADA

Gabriel Navarro ([gnavarro@ugr.es](mailto:gnavarro@ugr.es), [gnavarro@decsai.ugr.es](mailto:gnavarro@decsai.ugr.es))

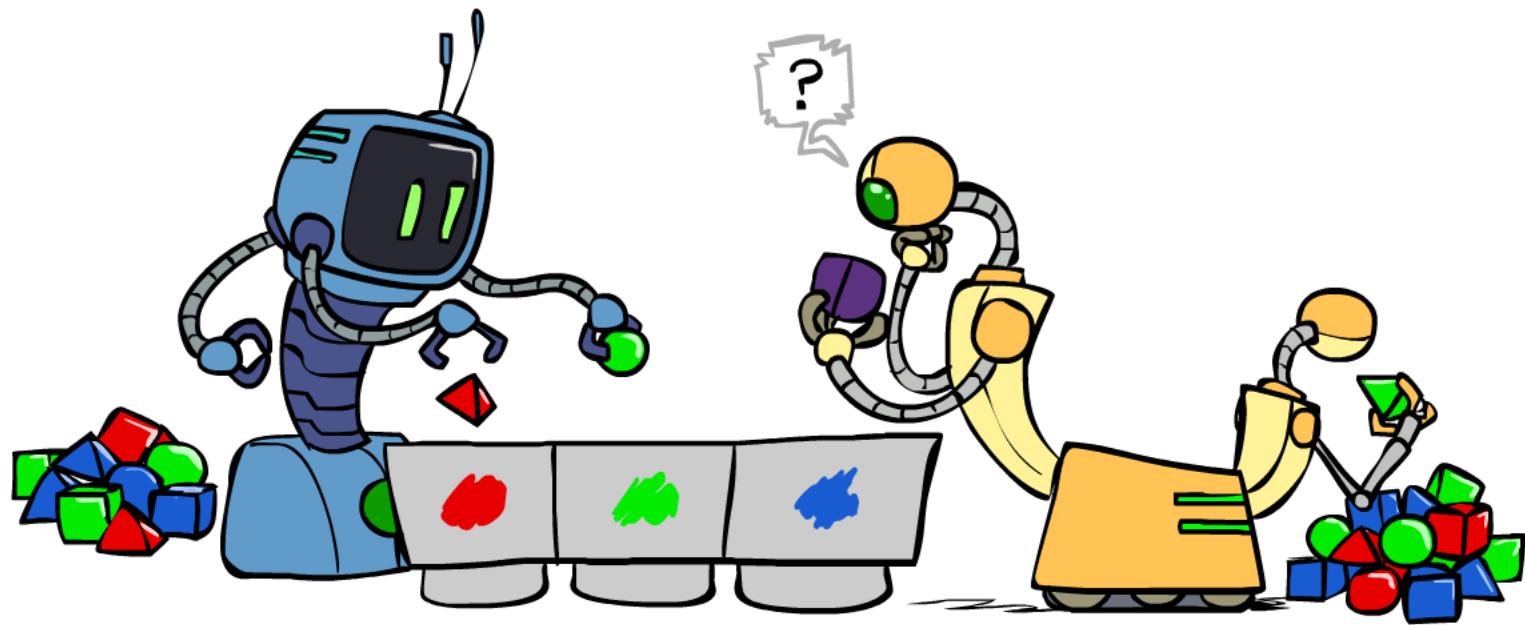
# Objetivos

- ❑ Entender el proceso de clustering como tarea descriptiva
- ❑ Conocer los modelos más usuales a la hora de realizar un agrupamiento y algunos algoritmos
- ❑ Conocer algunas medidas de evaluación del proceso de clustering

# Índice

- Concepto de clustering**
- Distancias
- Clustering jerárquico
- Clustering basado en representantes
- Clustering basado en densidad
- Grid-based methods
- Clustering basado en modelos
- Evaluación/validación del clustering

# Concepto de clustering

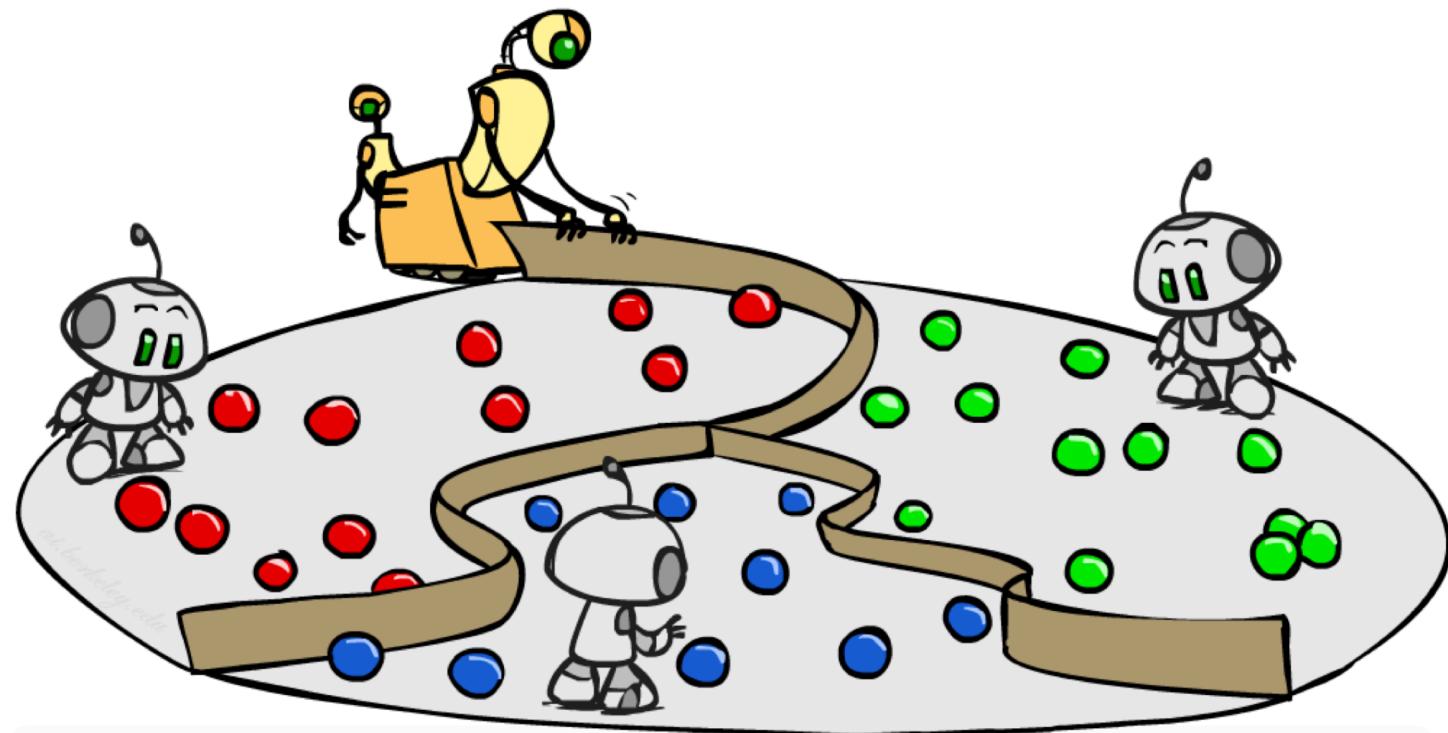


Es una tarea descriptiva

Es aprendizaje no supervisado

# Concepto de clustering

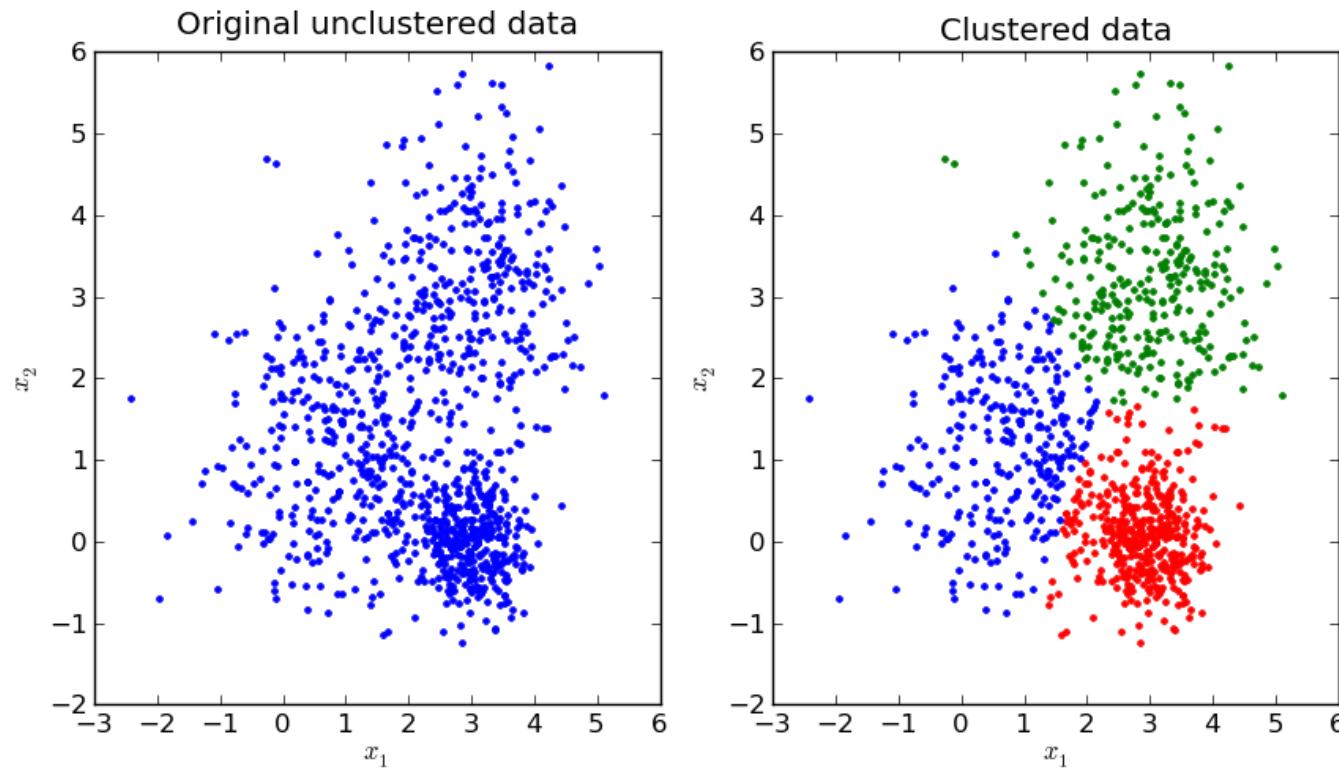
Idea básica Agrupar los elementos de un conjunto de manera que los elementos de cada grupo sean 'similares' y 'diferentes' de los objetos de los otros grupos



Normalmente, por proximidad

# Concepto de clustering

## Un ejemplo 2D



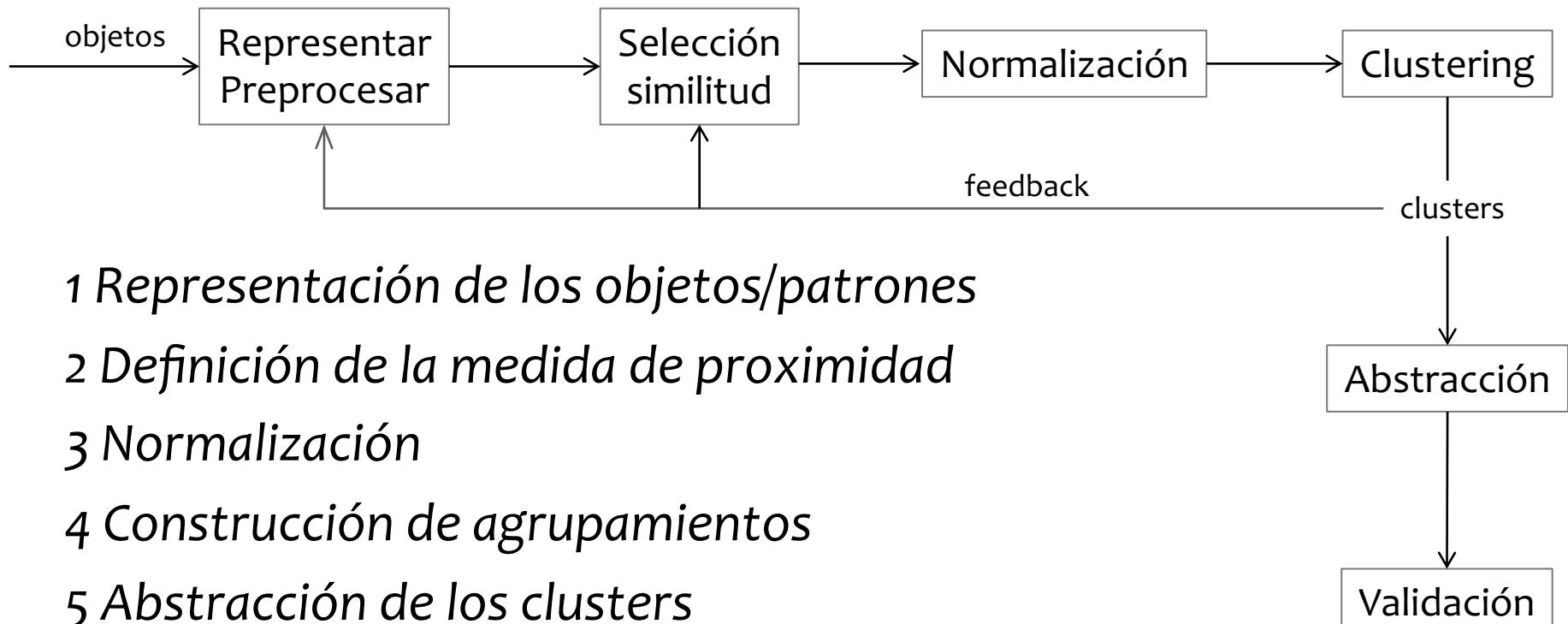
Los elementos del conjunto suelen estar representados por un vector, donde el valor de cada componente es la medida de una propiedad de los objetos

# Concepto de clustering

- ❑ Trata de obtener patrones de datos sin clasificar
- ❑ Es muy útil cuando no sabemos que estamos buscando
- ❑ Requiere datos, no etiquetas
- ❑ En muchas ocasiones, se obtienen resultados sin sentido
- ❑ Es complicado de evaluar o comparar distintos métodos

# Concepto de clustering

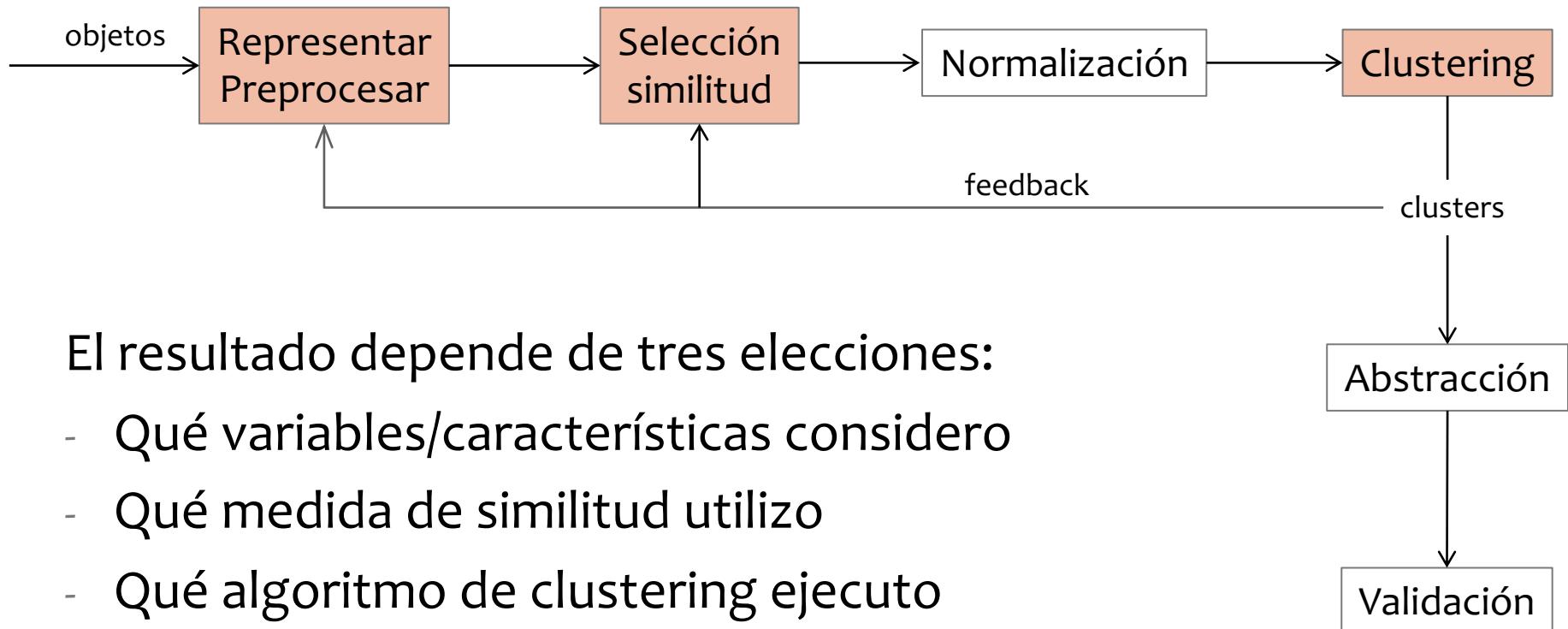
## Fases del proceso de clustering



- 1 Representación de los objetos/patrones
- 2 Definición de la medida de proximidad
- 3 Normalización
- 4 Construcción de agrupamientos
- 5 Abstracción de los clusters
- 6 Validación

# Concepto de clustering

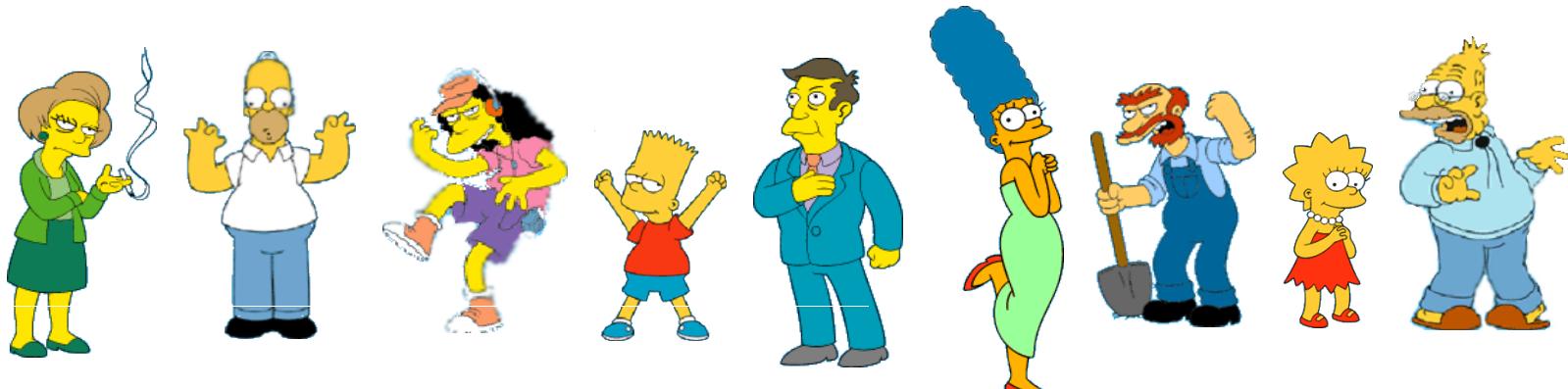
## Fases del proceso de clustering



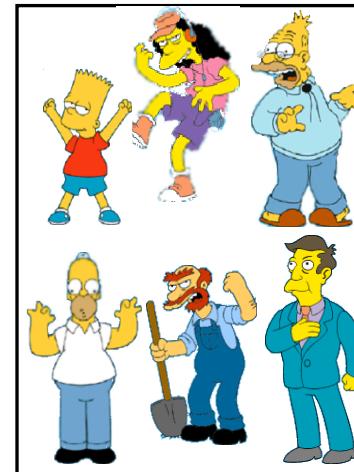
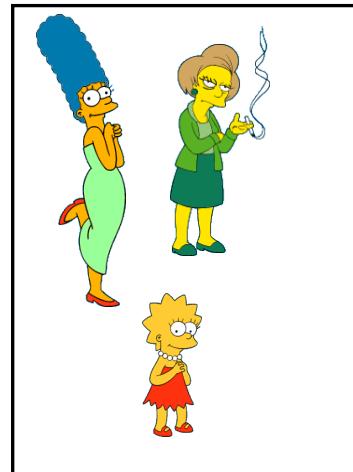
# Concepto de clustering

El proceso de clustering es subjetivo

¿Cuál es la forma natural de agrupar los personajes?



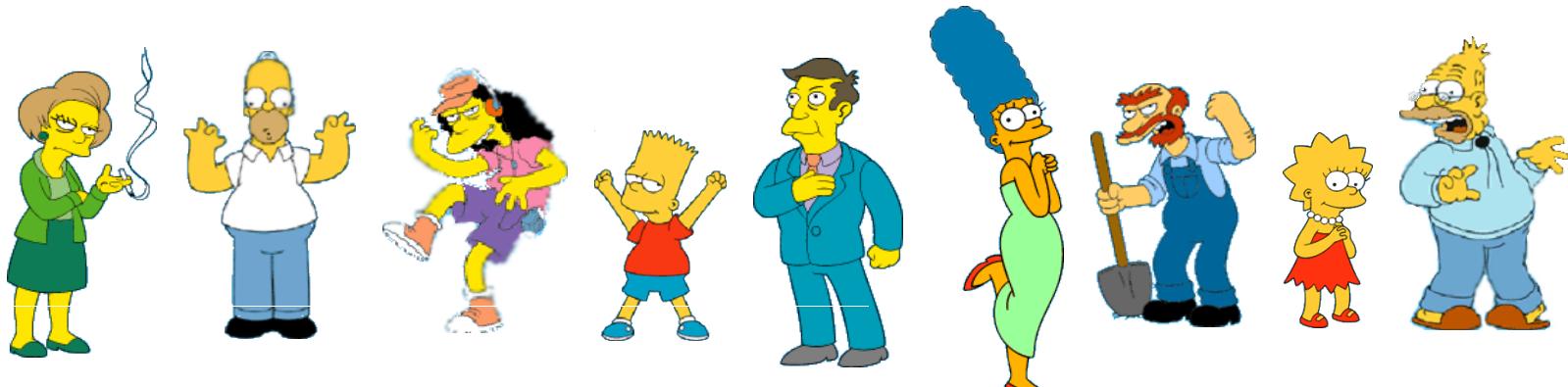
Hombres  
vs.  
Mujeres



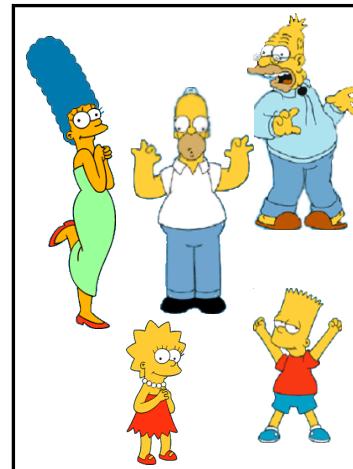
# Concepto de clustering

El proceso de clustering es subjetivo

¿Cuál es la forma natural de agrupar los personajes?

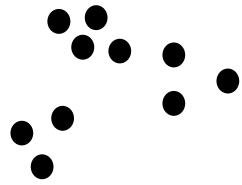


Simpsons  
vs.  
Empleados  
de la escuela  
de Springfield

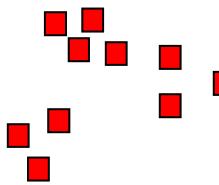
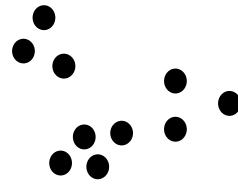


# Concepto de clustering

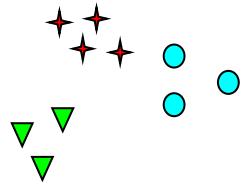
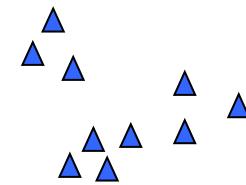
El proceso de clustering es subjetivo



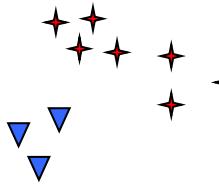
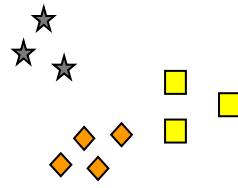
¿Cuántos  
agrupamientos?



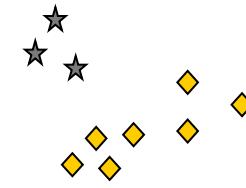
¿Dos?



¿Seis?



¿Cuatro?



# Concepto de clustering

¿Para qué queremos esto?

Por ejemplo, compresión de imágenes con perdida

$K = 2$



$K = 3$



$K = 10$

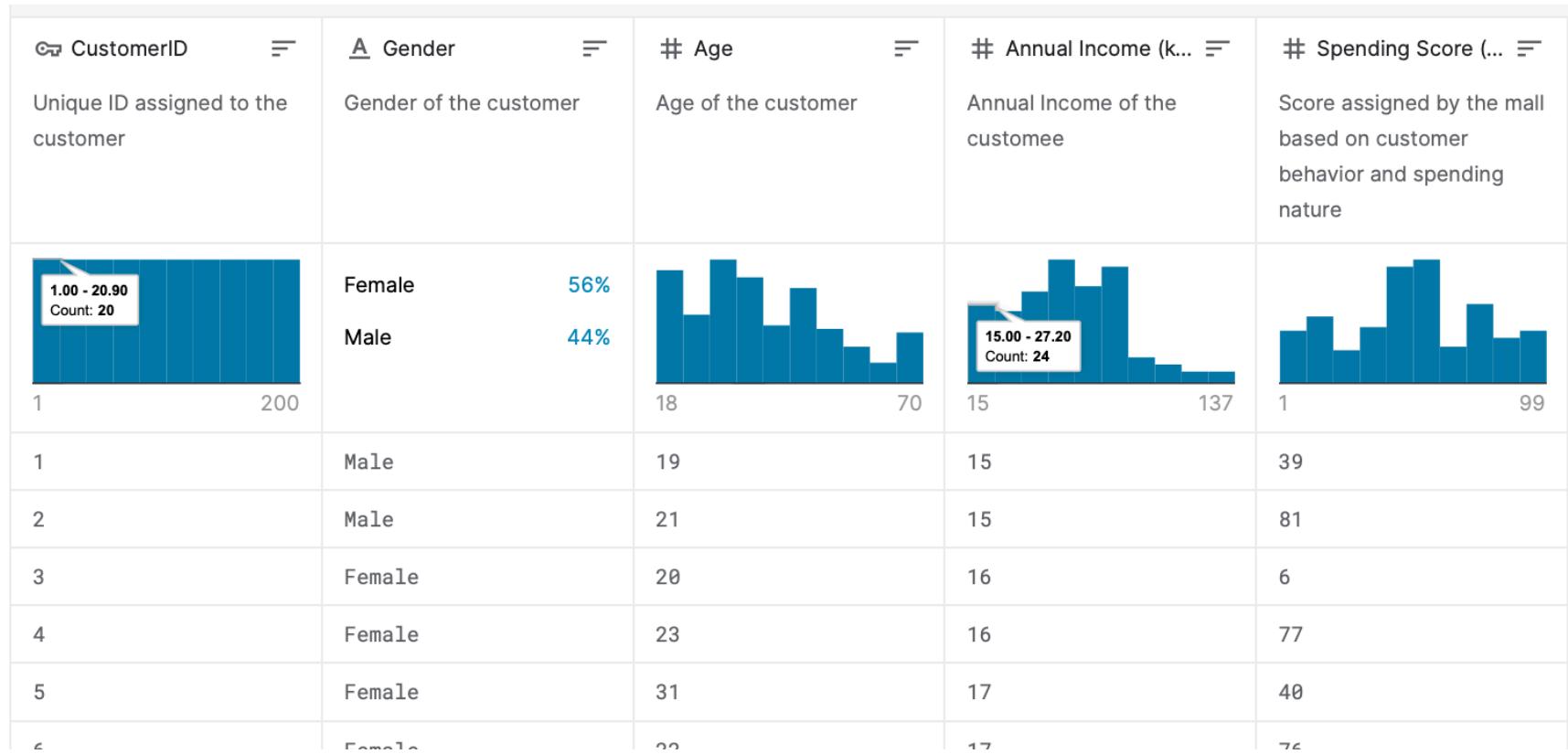


Original image



# Concepto de clustering

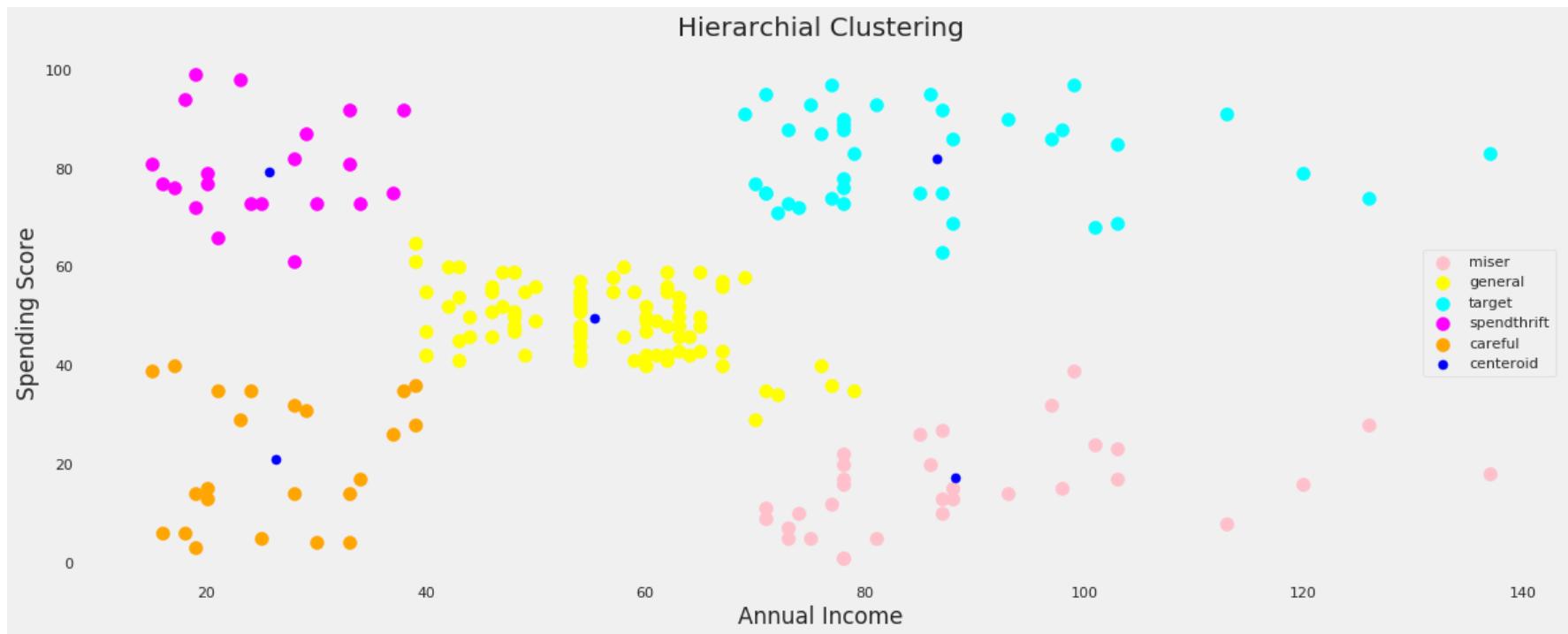
## Agrupamiento de clientes (*customer-segmentation*)



<https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python>

# Concepto de clustering

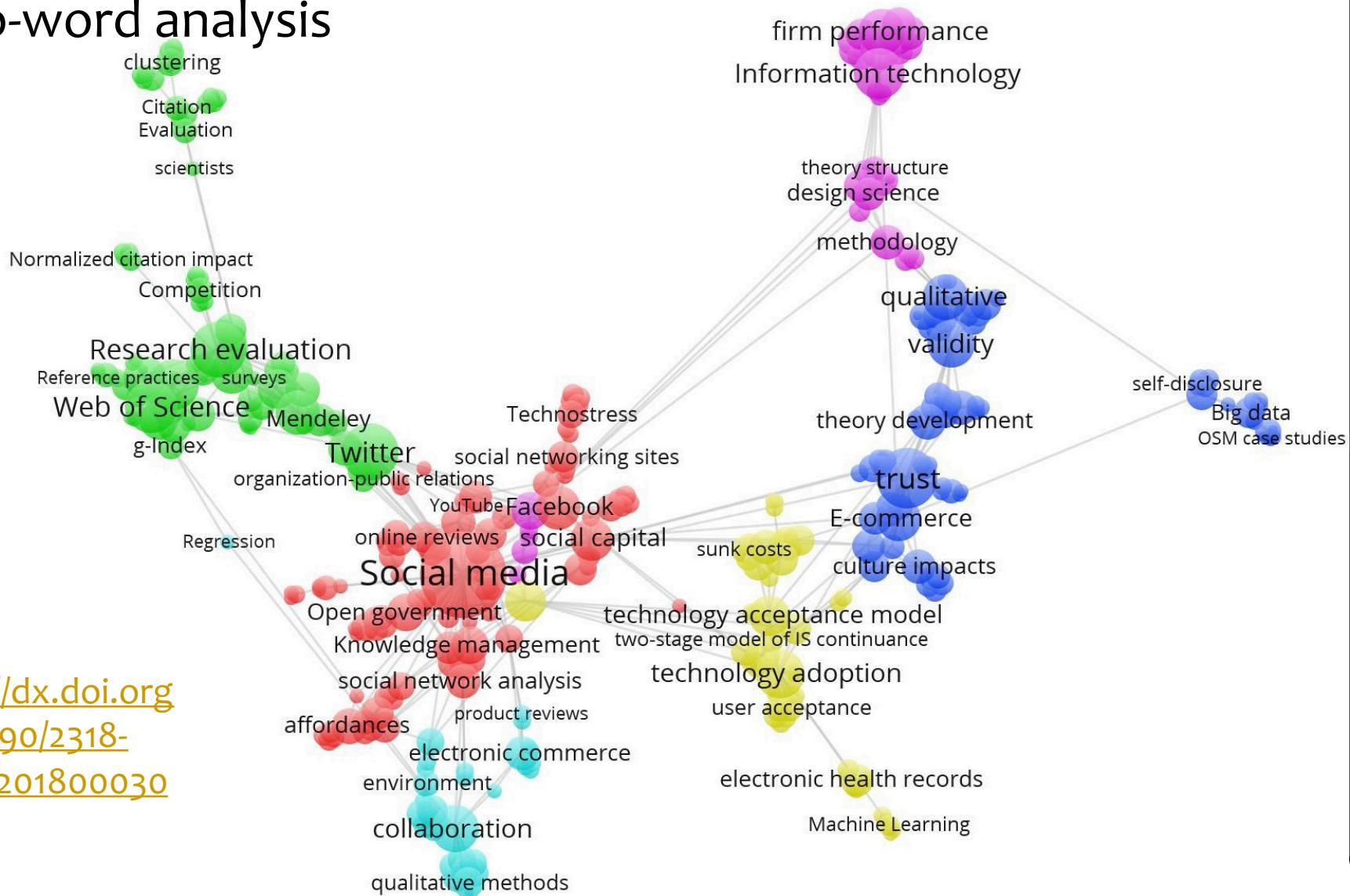
Agrupamiento de clientes (*customer-segmentation*)



<https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python>

# Concepto de clustering

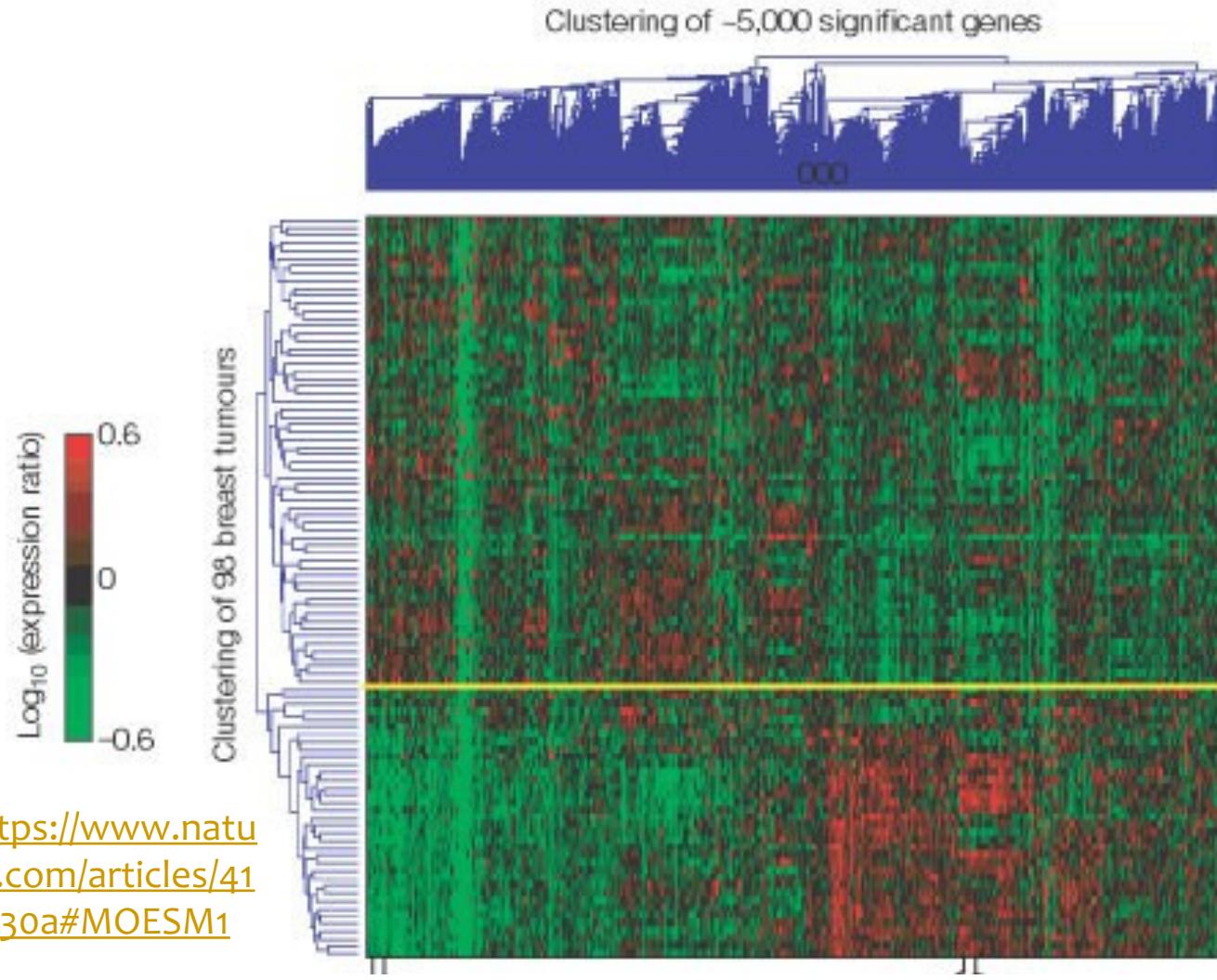
## Co-word analysis



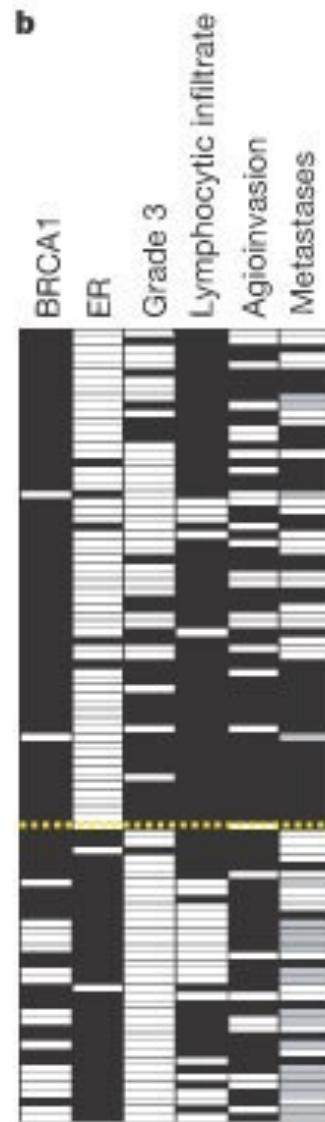
- Cluster 1: "Medios de comunicación sociales" (26%). Agrupó un total de 96 ítems, las palabras clave con mayor peso fueron: *Social media; Facebook; social capital; Knowledge management; Open government; affordances; user-generated content; social network analysis; text mining; knowledge sharing; Social Networks; blog; Communication; online reviews*.
- Cluster 2: "Evaluación de la investigación y nuevos indicadores basados en la Web 2.0" (22,7%). Incluyó un total de 84 ítems, entre las palabras clave con mayor peso se situaron: *Twitter; Research evaluation; Web of Science; Citation análisis; Altmetrics; Bibliometrics; Scopus; Webometrics; Google scholar; H-index; Online reference managers; Microblogging; Mendeley; Scientometrics; University rankings; Database coverage; Research metrics*.
- Cluster 3: "Confianza en los entornos virtuales" (19,6%). Integró un total de 72 ítems, las palabras clave con mayor peso: *trust; validity; qualitative; E-commerce; theory development; qualitative análisis; institutional mechanisms; Satisfaction; Big data; privacy; credibility; e-loyalty; Social comerse*.
- Cluster 4: "Modelo de aceptación de las Tecnologías de la Información (TIs)" (14%). Agrupó un total de 52 ítems, las palabras clave con mayor peso: *technology adoption; E-government; technology acceptance model; Unified Theory of Acceptance and Use of Technology (UTAUT); structural equation modelling; electronic health records*.
- Cluster 5: "Tecnología de la Información (TI) y resultados de las empresas" (9,3%). Incluyó un total de 34 ítems, las palabras clave con mayor peso: *Information technology; firm performance; resource-based view; revenue growth; cost reduction; profitability; design science; design theory; information systems*.
- Cluster 6: "Plataformas de comercio electrónico" (8,4%). Agrupó un total de 31 ítems, las palabras clave con mayor peso: *collaboration; electronic commerse; platforms; architecture; qualitative methods; virtual teams; innovation; outsourcing; offshore software development; environment*.

# Concepto de clustering

## a Gene expression analysis



<https://www.nature.com/articles/415530a#MOESM1>



# Concepto de clustering

Algunas aplicaciones más:

- ❑ Information retrieval
  - documents/multimedia data
- ❑ Internet security
  - fraud/spam detection
- ❑ Social network discovery
- ❑ Finance, marketing, insurance, banking
  - Discovering distinct groups of customers
- ❑ Reconocimiento de formas
- ❑ ...

# Concepto de clustering

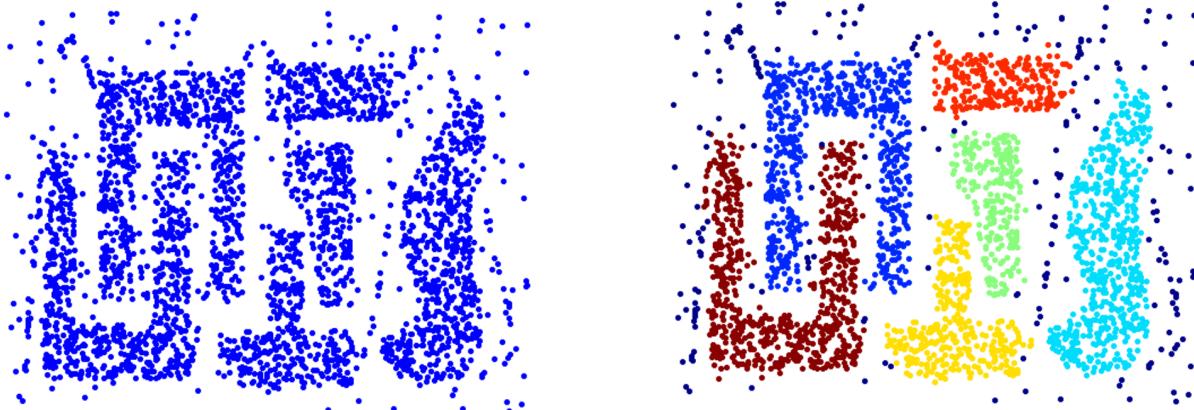
En general, se busca un mayor entendimiento de los datos

Concretamente, también puede ayudar a:

Reducción de características

Selección de prototipos (cluster sampling)

Detección de anomalías (outliers)

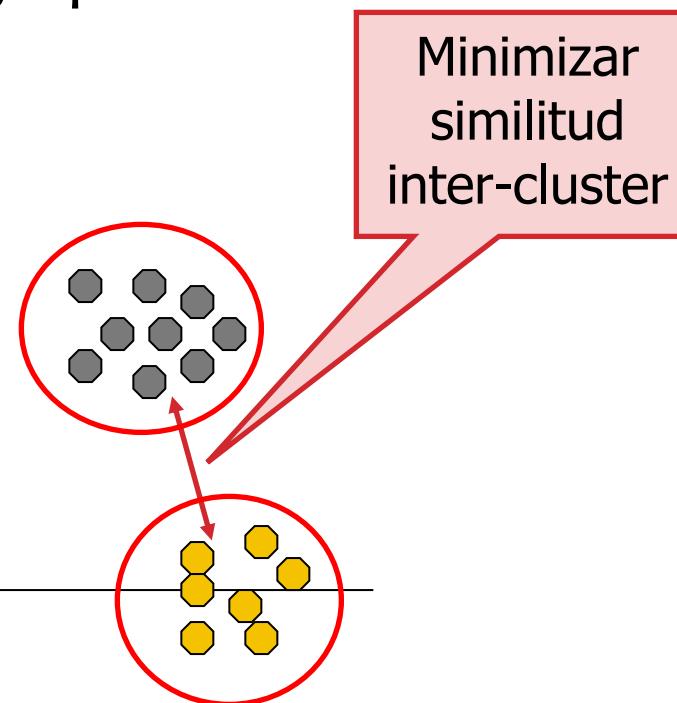
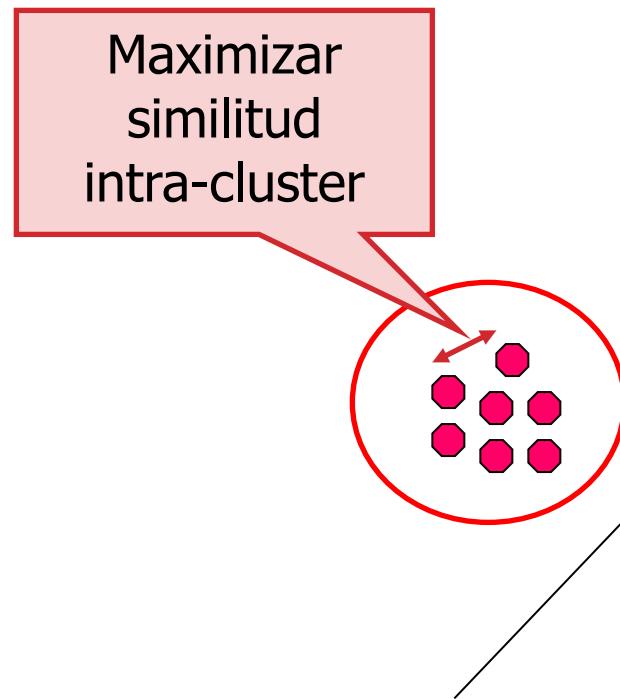


# Índice

- Concepto de clustering
- Distancias**
- Clustering jerárquico
- Clustering basado en representantes
- Clustering basado en densidad
- Grid-based methods
- Clustering basado en modelos
- Evaluación/validación del clustering

# Similitud

Se busca encontrar agrupamientos de tal forma que los objetos de un grupo sean **similares** entre sí y diferentes de los objetos de otros grupos



# Distancia

La similitud normalmente viene determinada por la distancia entre puntos.

Dado un conjunto  $X$ , una distancia (o métrica) en dicho conjunto es una función

$$d : X \times X \longrightarrow \mathbb{R}$$

verificando las siguientes propiedades:

- Propiedad reflexiva  $d(a, b) = 0 \iff a = b$
- Propiedad simétrica  $d(a, b) = d(b, a) \quad \forall a, b$
- Desigualdad triangular  $d(a, b) \leq d(a, c) + d(c, b) \quad \forall a, b, c$

**A mayor distancia, menor similitud. Y viceversa.**

# Distancia

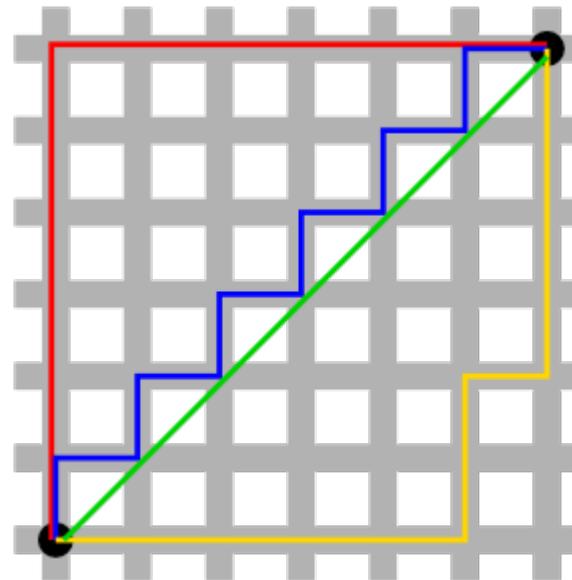
**¿qué distancia utilizamos?** Depende de la distribución de los datos, la dimensionalidad y el tipo de los datos

## Datos numéricos

- Euclídea       $d_e(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$
- Manhattan       $d_M(x, y) = \sum_i |x_i - y_i|$
- Minkowski       $d^k(x, y) = \left( \sum_i |x_i - y_i|^k \right)^{\frac{1}{k}}$
- Chebyshev       $d_\infty(x, y) = \max_j |x_j - y_j|$

# Distancia

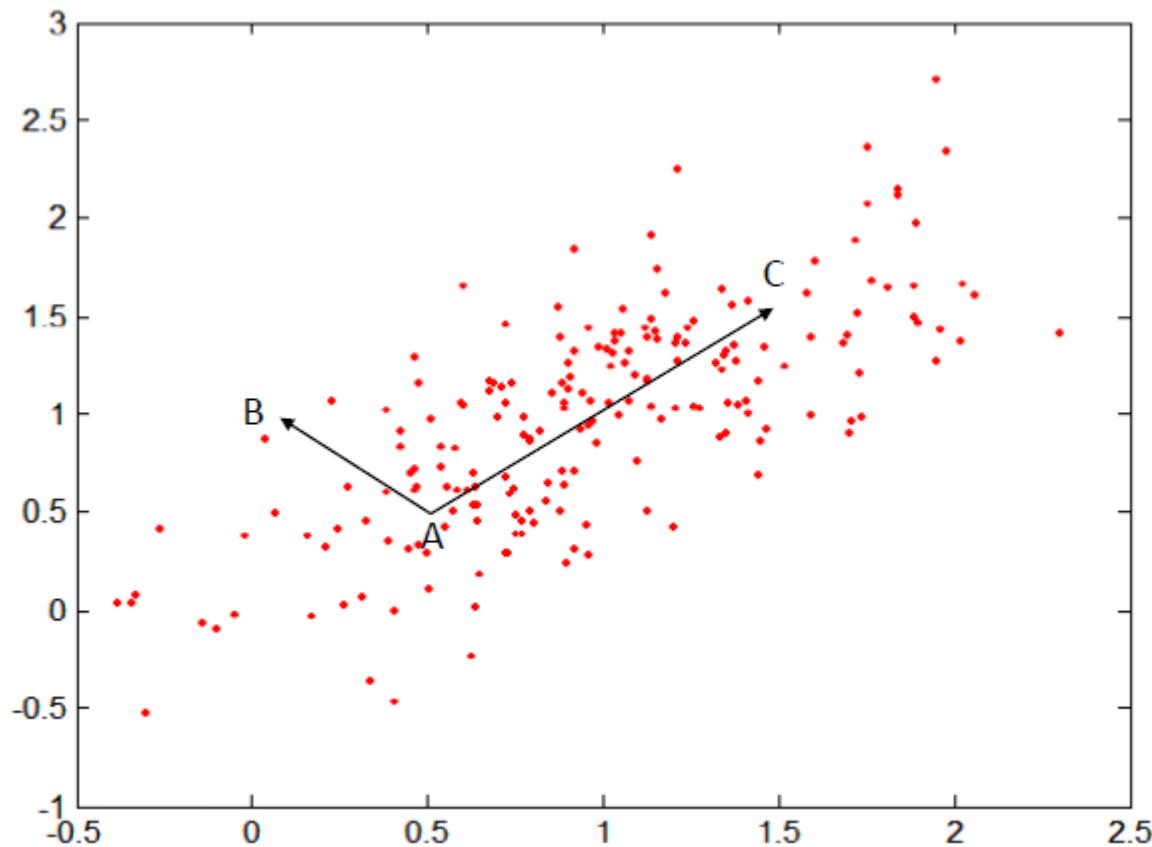
- Euclídea = 8.48
- Manhattan = 12
- Chebyshev = 6



# Distancias

## □ Distancia de Mahalanobis

$$d(x, y) = (x - y)\Sigma^{-1}(x - y)^T \text{ con } \Sigma \text{ matriz de covarianzas}$$



$$\Sigma = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$$

A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

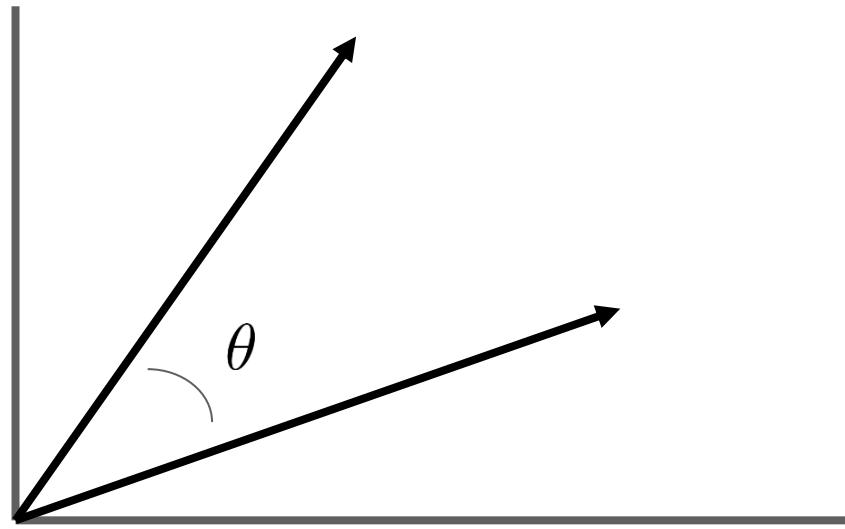
Mahal(A,B) = 5

Mahal(A,C) = 4

# Distancias

## ❑ Coseno del ángulo de los vectores

$$d(x, y) = \cos \theta = \frac{\sum_i x_i \cdot y_i}{|x||y|}$$



# Distancias

## Datos binarios/discretos

Hamming       $d_H(x, y) = \#\{i \text{ tales que } x_i \neq y_i\}$

Distancia de Jaccard

$$J(A, B) = \frac{J_{01} + J_{10}}{J_{01} + J_{01} + J_{11}}$$

donde:

$J_{01} =$  # componentes con 0 en  $A$  y 1 en  $B$

$J_{10} =$  # componentes con 1 en  $A$  y 0 en  $B$

$J_{11} =$  # componentes con 1 en  $A$  y 1 en  $B$

# Distancias

## Cadenas

### Distancia de Levenshtein

Número mínimo de operaciones (insertar, eliminar, sustituir) para transformar una cadena en otra

$$d(\text{casa}, \text{pasa}) = 1$$

$$d(\text{casa}, \text{casas}) = 1$$

$$d(\text{casa}, \text{asa}) = 1$$

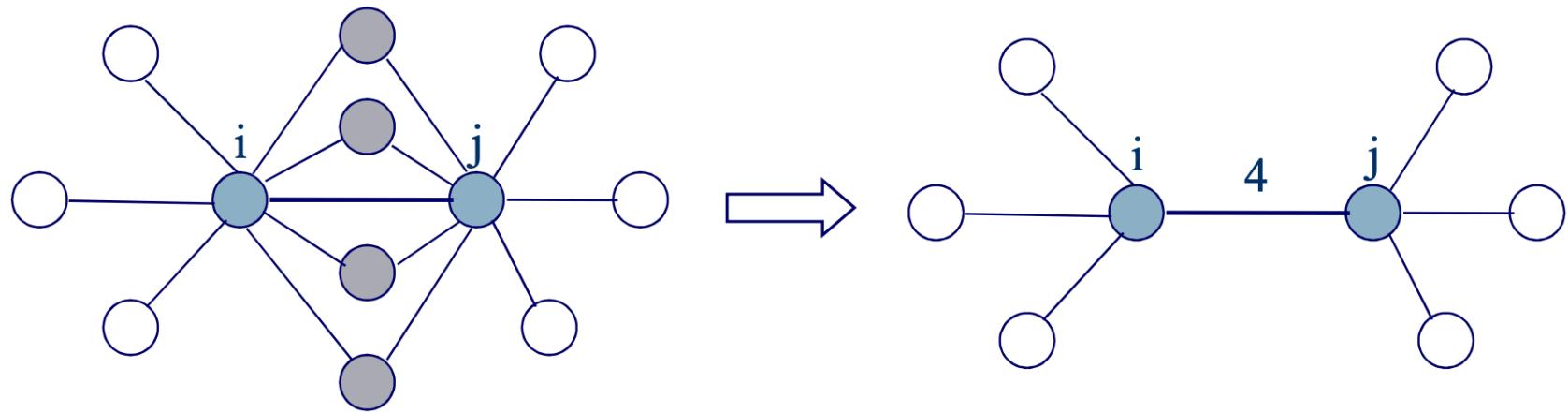
Se calcula mediante programación dinámica.

Aplicaciones: Correctores ortográficos, reconocimiento de voz, detección de plagios, análisis de ADN

# Distancias

## Grafos

- Similitud según vecinos comunes



# Tipos de clustering

Requisitos del algoritmo “perfecto”

- Escalabilidad
- Manejo de distintos tipos de datos
- Identificación de clusters con formas arbitrarias
- Número mínimo de parámetros
- Tolerancia frente a ruido y outliers
- Independencia con respecto al orden de presentación de los patrones de entrenamiento
- Posibilidad de trabajar en espacios con muchas dimensiones diferentes
- Capacidad de incorporar restricciones especificadas por el usuario (“domain knowledge”)
- Interpretabilidad / Usabilidad

# Tipos de clustering

- Según el tipo de agrupamiento
  - **Jerárquico**, establece una jerarquía en los datos
  - **Particional**, divide el conjunto en subgrupos
- En particionales, según la forma de construir los clusters
  - **Representative-based**, busca regiones cercanas a unos representantes/puntos del conjunto de datos
  - **Density-based**, busca regiones densas
  - **Grid-based**, dividen el espacio en una malla
  - **Model-based**, supone que los datos se distribuyen según modelo (normalmente, probabilístico)
  - **Graph-based**, construye un grafo con los datos
  - **Search-based**, supone el clustering como un problema de optimización
  - ...

# Tipos de clustering

- ❑ Según la pertenencia a los clusters
  - **Exclusivo**, cada elemento sólo a un cluster
  - **Solapado**, un elemento en varios cluster
  - **Difuso**, tienen un grado de pertenencia a los cluster
- ❑ Según los elementos que se agrupan
  - **Completa**, todos los elementos en algún cluster
  - **Parcial**, algunos elementos sin cluster
- ❑ Según si hay dependencia del tiempo
  - **Estático**, ni los clusters ni los datos cambian
  - **Dinámico**, cambian los clusters y/o los datos

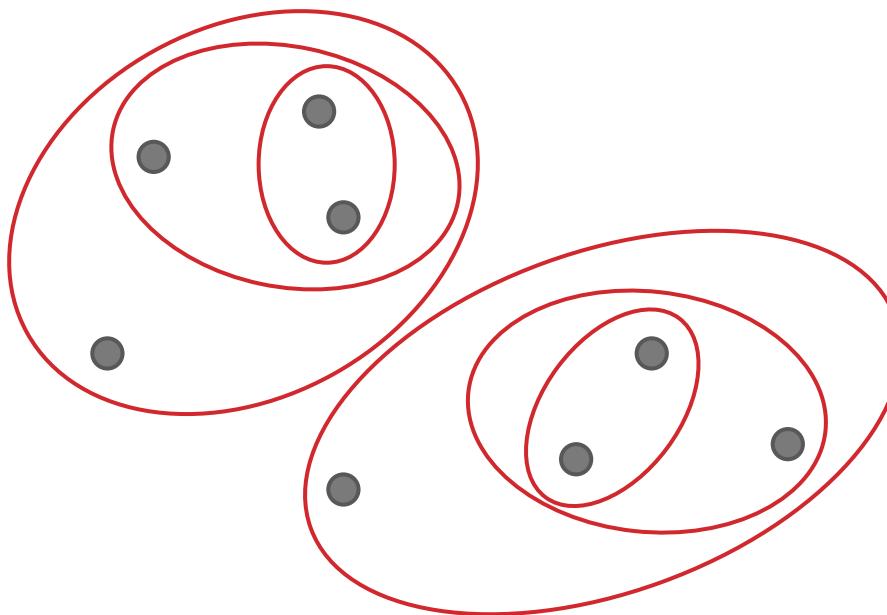
Seguramente haya más clasificaciones...

# Índice

- Concepto de clustering
- Distancias
- Clustering jerárquico**
- Clustering basado en representantes
- Clustering basado en densidad
- Grid-based methods
- Clustering basado en modelos
- Evaluación/validación del clustering

# Clustering jerárquico

Consiste en crear una jerarquía en los datos de forma que se crean clusters anidados, es decir, particiones crecientes/decrecientes del espacio de datos

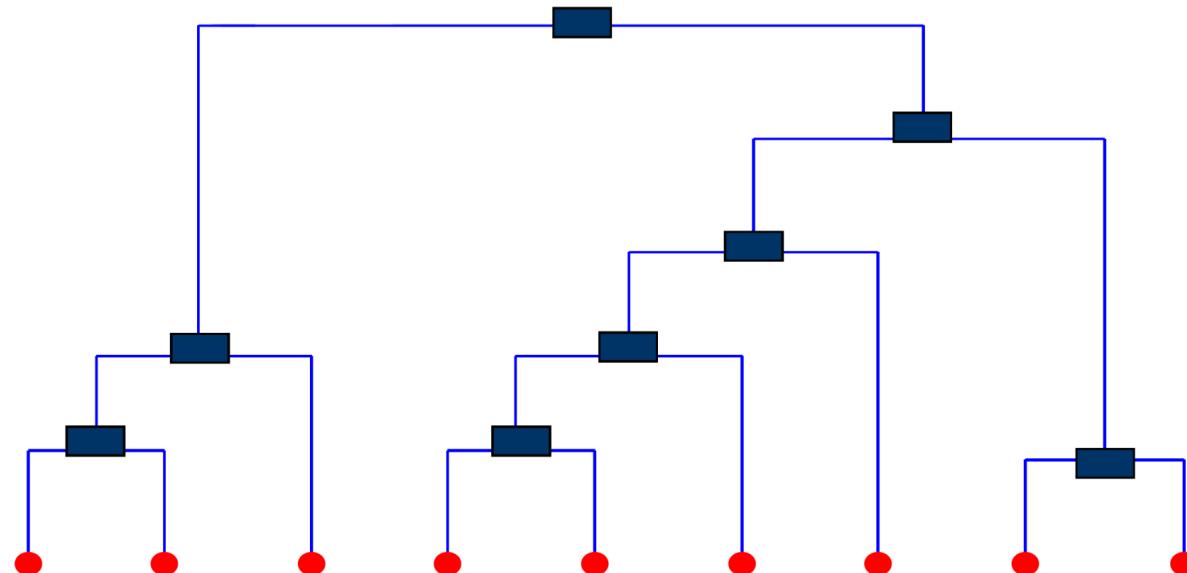


Podemos pensar  
como cadenas de  
relaciones de  
equivalencia!

Para representar las particiones se suelen utilizar **dendrogramas**

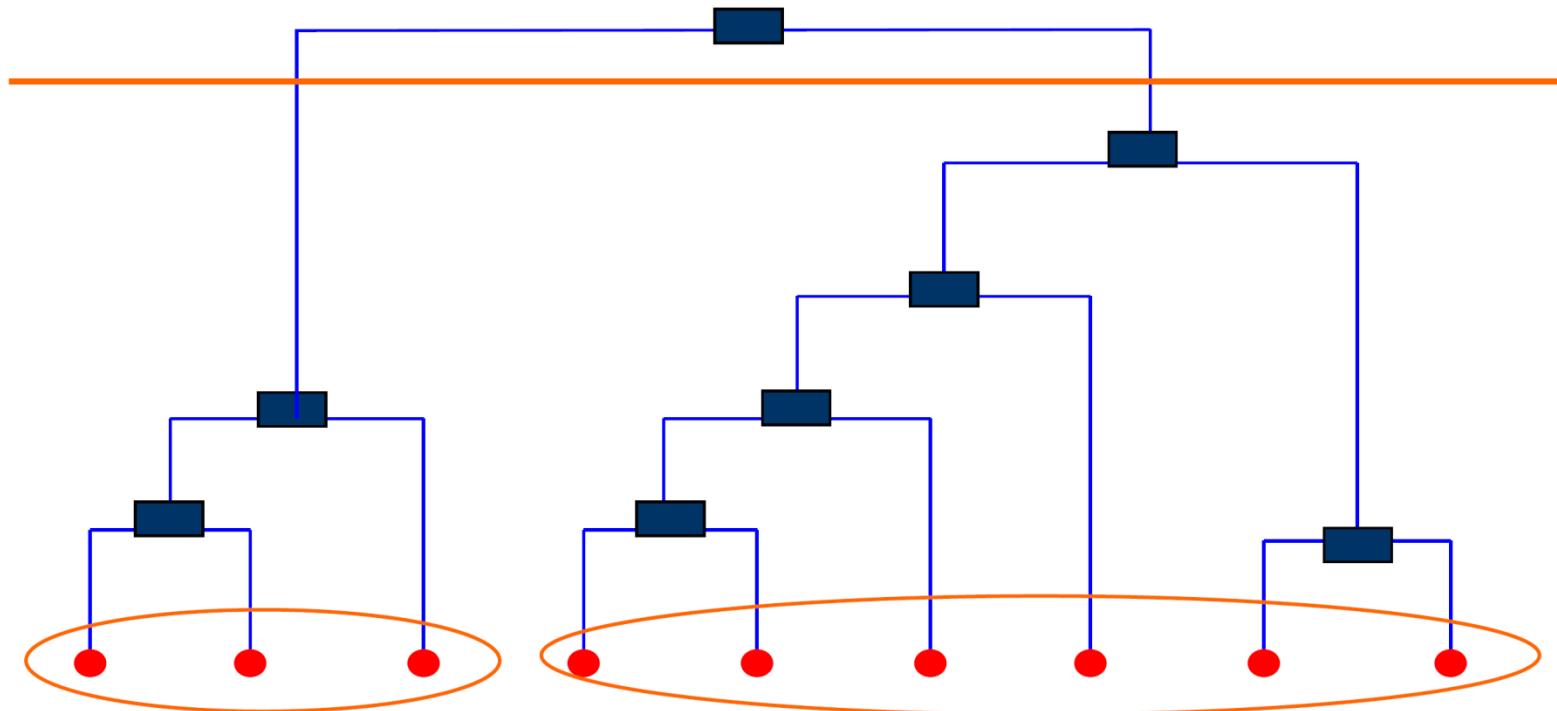
# Dendrogramas

- Un **dendrograma** es un árbol que muestra como los clusters se mezclan/escinden jerárquicamente
- Cada nodo es un cluster, que contiene a los descendientes hoja. Cada hoja es un dato
- La similitud entre dos objetos viene dada por la “altura” del nodo común más cercano



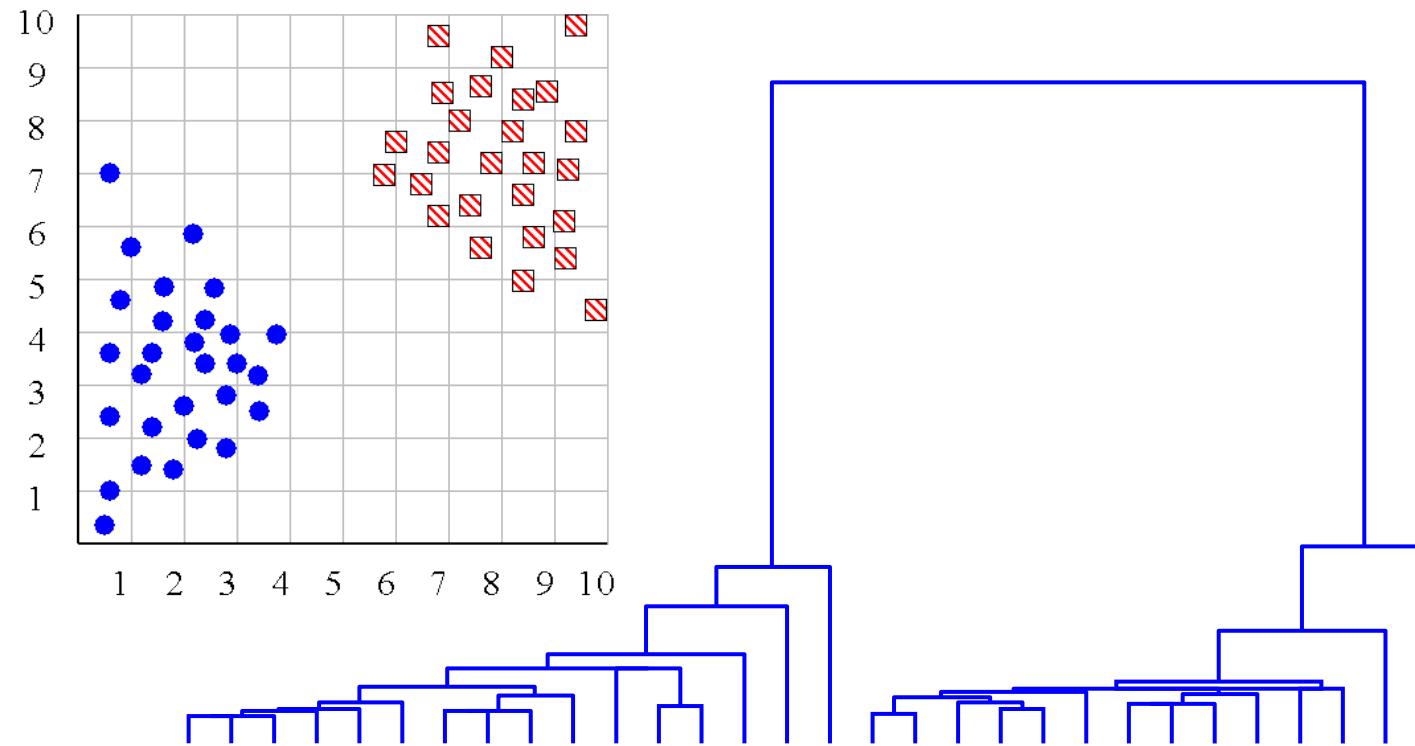
# Dendrogramas

Una partición en clusters se obtiene al cortar el dendrograma al nivel deseado. Cada componente conexa forma un cluster



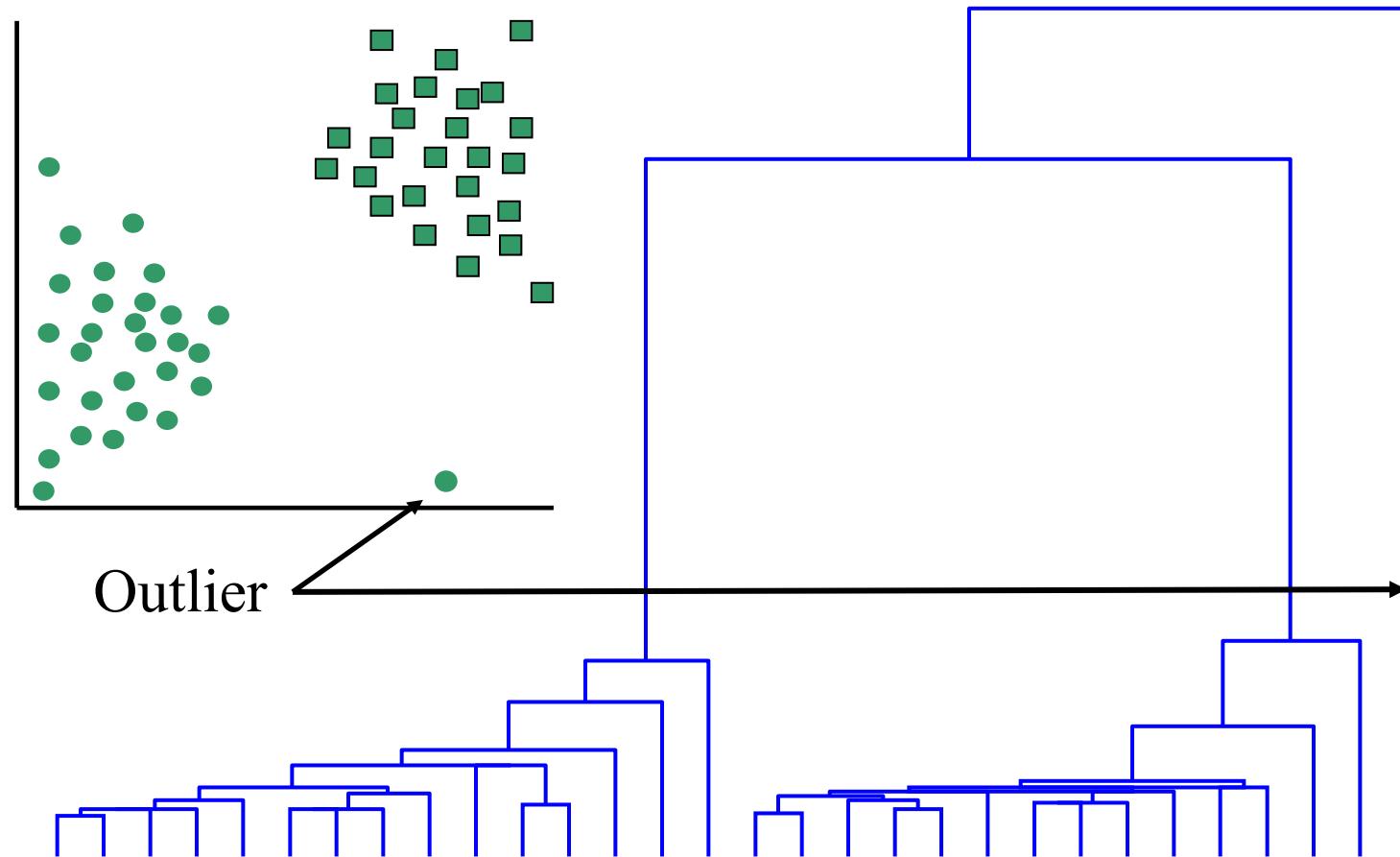
# Dendrogramas

Nos puede ayudar a determinar el número adecuado de agrupamientos (aunque normalmente no será tan fácil)

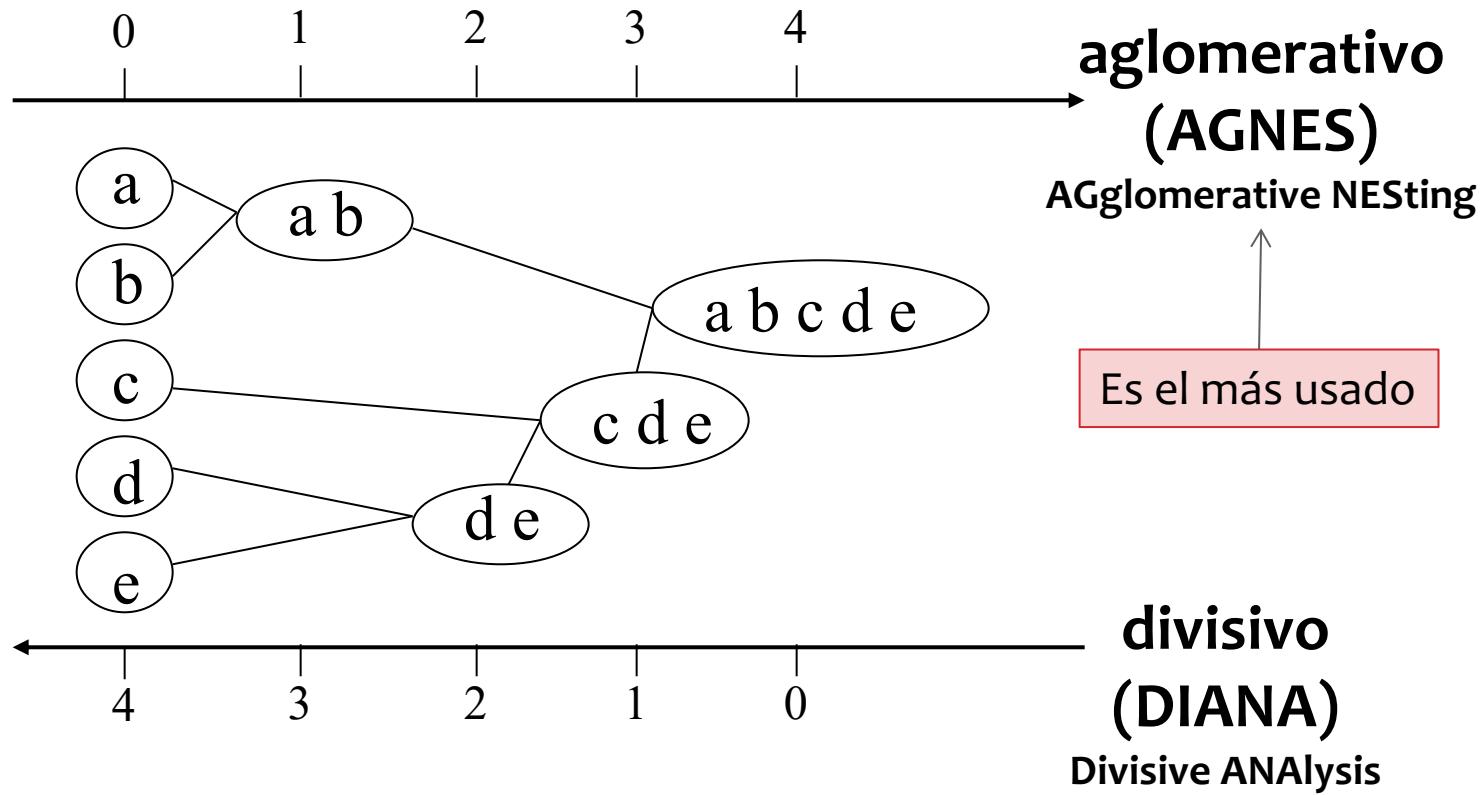


# Dendrogramas

Útil para la detección de outliers



# Enfoques



En lugar de establecer de antemano el número de clusters, tenemos que definir un **criterio de parada**

# Algoritmo básico aglomerativo

---

**Algorithm** Algoritmo básico aglomerativo

---

**Input:**  $C = \{c_i\}_{i=1,\dots,m}$  ejemplos,  $d$  distancia

**Output:** Dendrograma  $D$

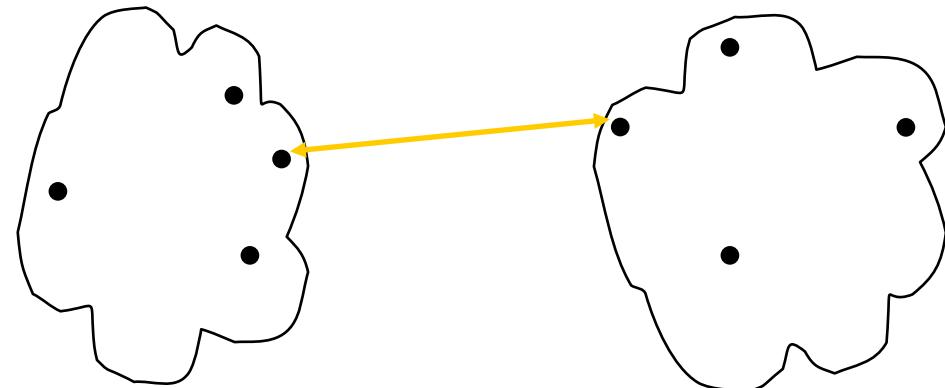
- 1: Calcular matriz de distancias  $M$
  - 2: Asignar cada ejemplo  $c_i$  a un cluster  $D_i$
  - 3: **while** Exista más de un cluster **do**
  - 4:     Se busca el par de clusters más cercanos  $(D_i, D_j)$
  - 5:     Se fusionan en un cluster  $D_{ij}$
  - 6:     Se calcula la distancia de  $D_{ij}$  al resto de clusters
  - 7: **return** El árbol  $D = \{D_\alpha\}$
- 

¿Cuál es la distancia entre clusters?

# ¿Cómo medir la distancia entre clusters?

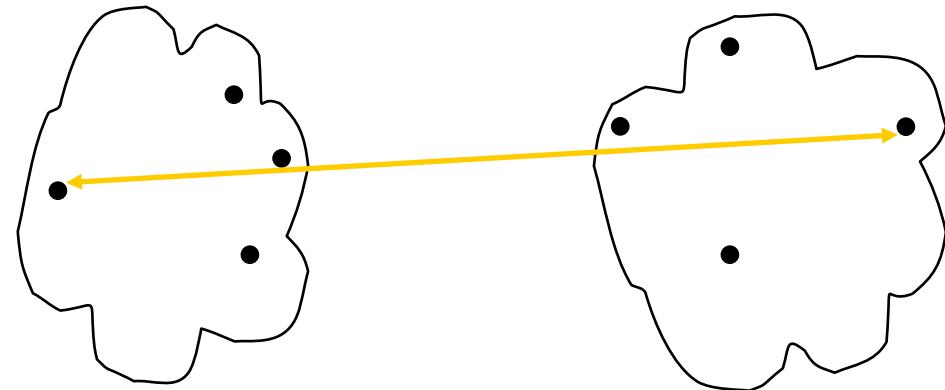
## MIN

Distancia mínima entre un punto de un cluster y otro del otro (single-link)



## MAX

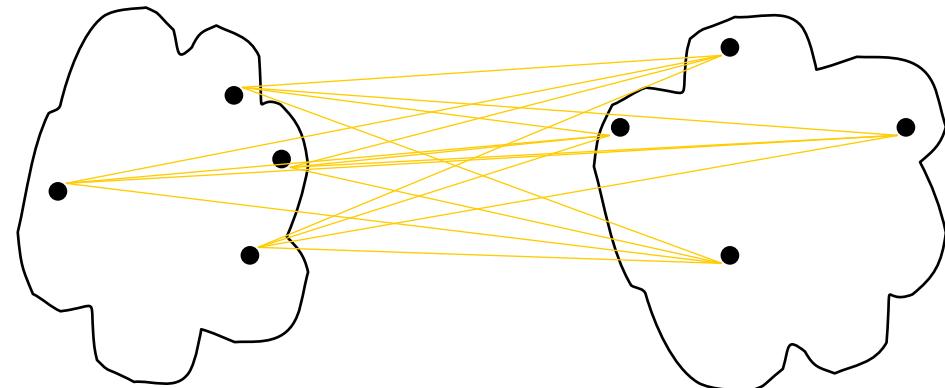
Distancia máxima entre un punto de un cluster y otro del otro (complete-link)



# ¿Cómo medir la distancia entre clusters?

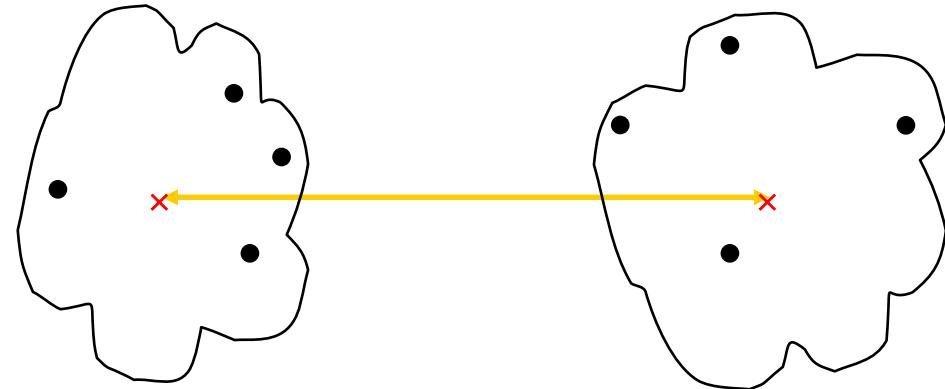
## Promedio

Distancia media entre un punto de un cluster y otro del otro



## Centróide

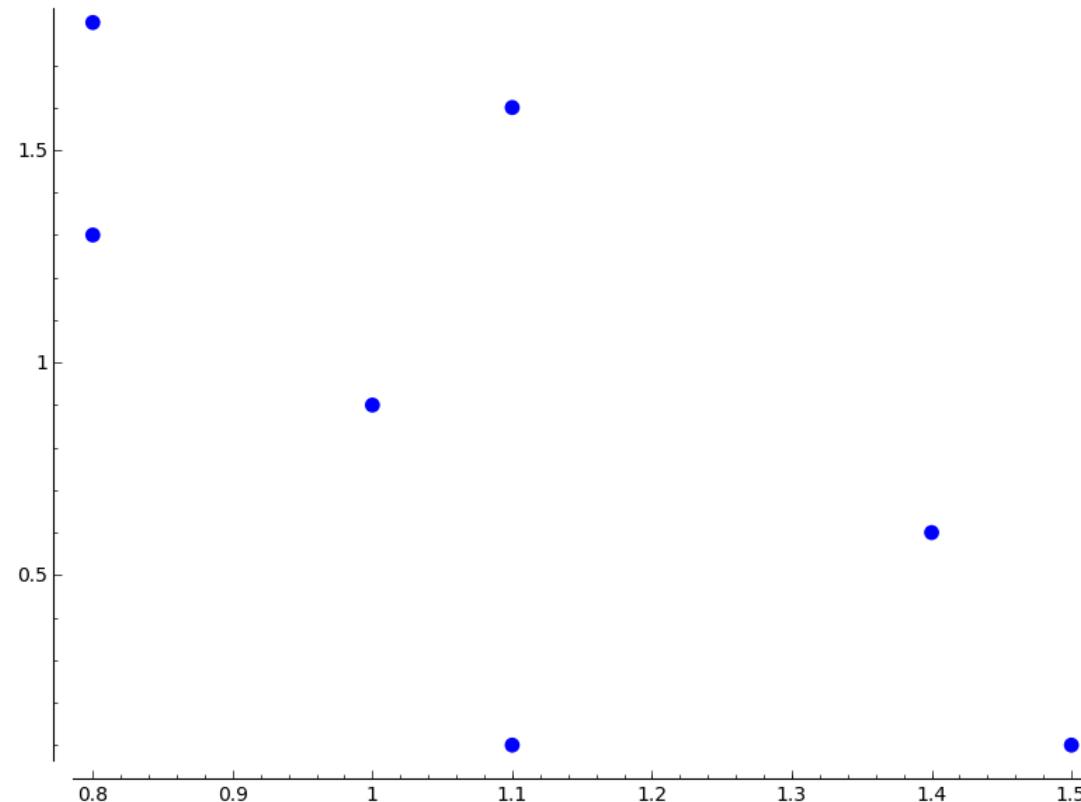
Distancia entre los centróides de los clusters



# Ejemplo

Consideremos los datos

$[[0.8, 1.8], [1.1, 1.6], [0.8, 1.3], [1.0, 0.9], [1.4, 0.6], [1.5, 0.1], [1.1, 0.1]]$



# Ejemplo

$[[0.8,1.8],[1.1,1.6],[0.8,1.3],[1.0,0.9],[1.4,0.6],[1.5,0.1],[1.1,0.1]]$

□ Calculamos la matriz de distancias

- Será la mínima entre los puntos de los clusters

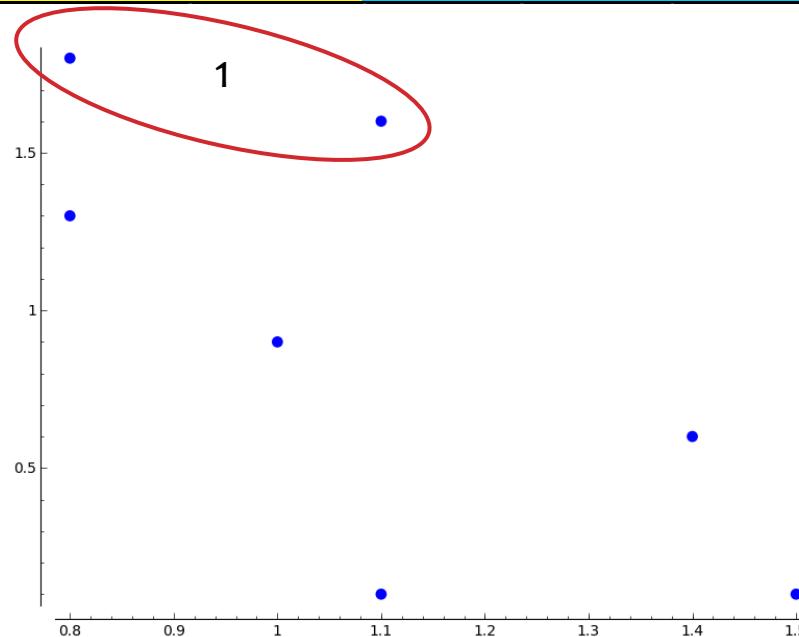
D1	D2	D3	D4	D5	D6	D7
D1	0.3606					
D2						
D3	0.5000	0.4243				
D4	0.9220	0.7071	0.4472			
D5	1.3416	1.0440	0.9220	0.5000		
D6	1.8385	1.5524	1.3892	0.9434	0.5099	
D7	1.7263	1.5000	1.2369	0.8062	0.5831	0.4000

# Ejemplo

$[[0.8, 1.8], [1.1, 1.6], [0.8, 1.3], [1.0, 0.9], [1.4, 0.6], [1.5, 0.1], [1.1, 0.1]]$

- ❑ Fusionamos primer y segundo cluster

D1	D2	D3	D4	D5	D6	D7
D1	0.3606					
D2		0.5000	0.4243			
D3	0.9220	0.7071	0.4472			
D4	1.3416	1.0440	0.9220	0.5000		
D5	1.8385	1.5524	1.3892	0.9434	0.5099	
D6	1.7263	1.5000	1.2369	0.8062	0.5831	0.4000
D7						

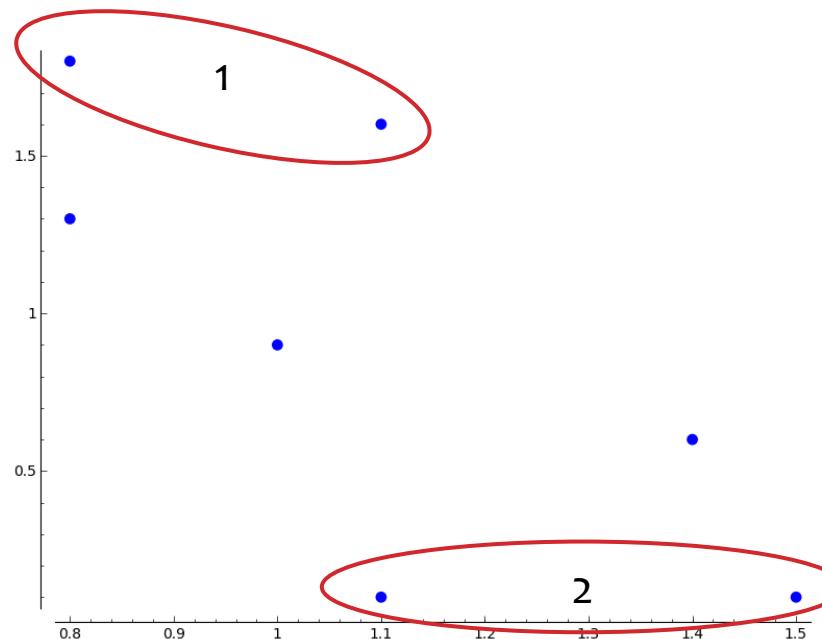


# Ejemplo

$[[0.8, 1.8], [1.1, 1.6], [0.8, 1.3], [1.0, 0.9], [1.4, 0.6], [1.5, 0.1], [1.1, 0.1]]$

- Recalculamos distancias y fusionamos clusters

D1/D2=D8	D3	D4	D5	D6	D7
D3	0.4243				
D4	0.7071	0.4472			
D5	1.3416	0.9220	0.5000		
D6	1.5524	1.3892	0.9434	0.5099	
D7	1.5000	1.2369	0.8062	0.5831	0.4000
D1/D2=D8	D3	D4	D5	D6	D7

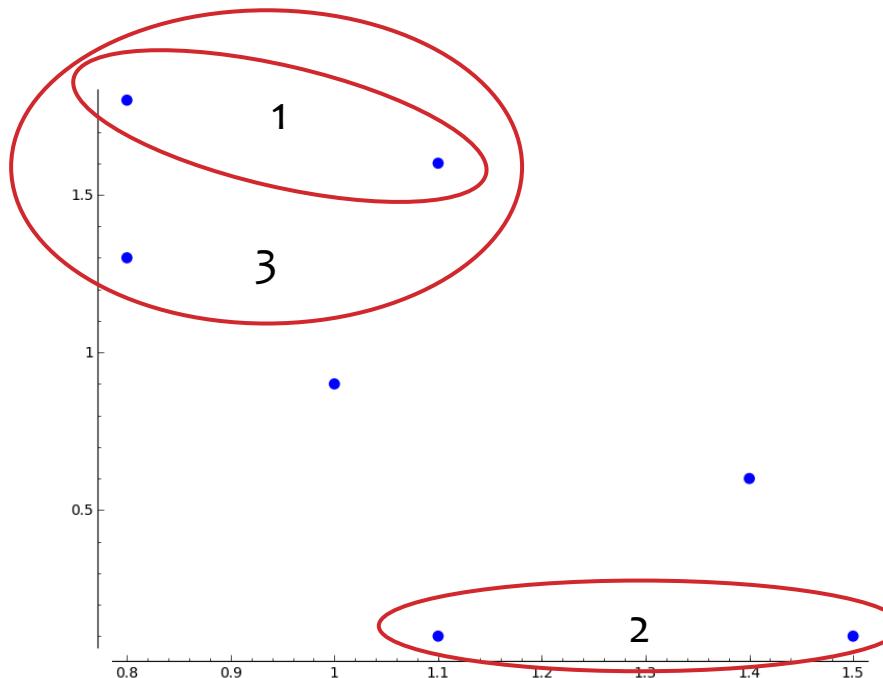


# Ejemplo

$[[0.8,1.8],[1.1,1.6],[0.8,1.3],[1.0,0.9],[1.4,0.6],[1.5,0.1],[1.1,0.1]]$

- Recalculamos distancias y fusionamos clusters

D8						
D3	0.4243					
D4	0.7071	0.4472				
D5	1.3416	0.9220	0.5000			
D6/D7=D9	1.5000	1.2369	0.8062	0.5831		
	D8	D3	D4	D5	D6/D7=D9	

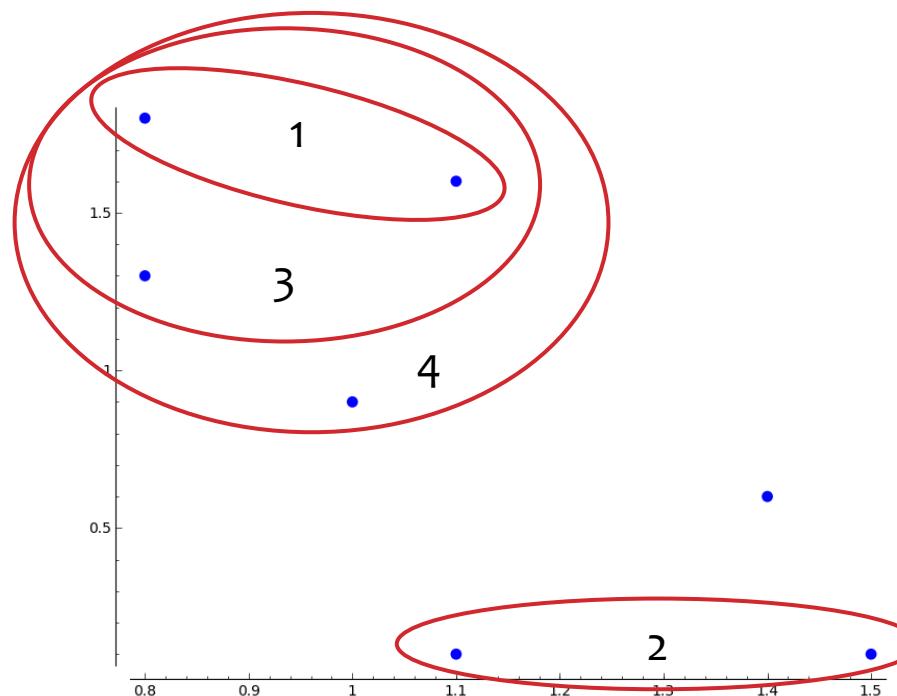


# Ejemplo

$[[0.8,1.8],[1.1,1.6],[0.8,1.3],[1.0,0.9],[1.4,0.6],[1.5,0.1],[1.1,0.1]]$

- Recalculamos distancias y fusionamos clusters

D3/D8=D10	D4	0.4472				
D5	0.9220	0.5000				
D9	1.2369	0.8062		0.5831		
	D3/D8=D10	D4		D5	D9	

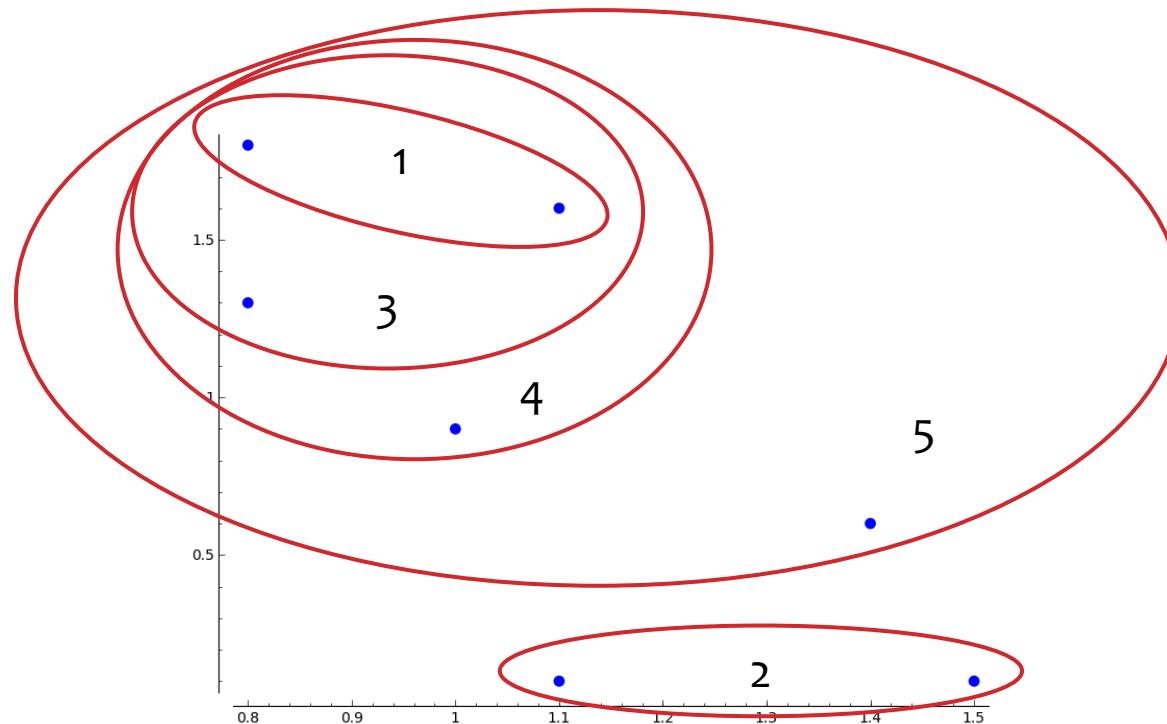


# Ejemplo

$[[0.8,1.8],[1.1,1.6],[0.8,1.3],[1.0,0.9],[1.4,0.6],[1.5,0.1],[1.1,0.1]]$

- Recalculamos distancias y fusionamos clusters

D4/D10=D11	
D5	0.5000
D9	0.8062      0.5831
D4/D10=D11	D5

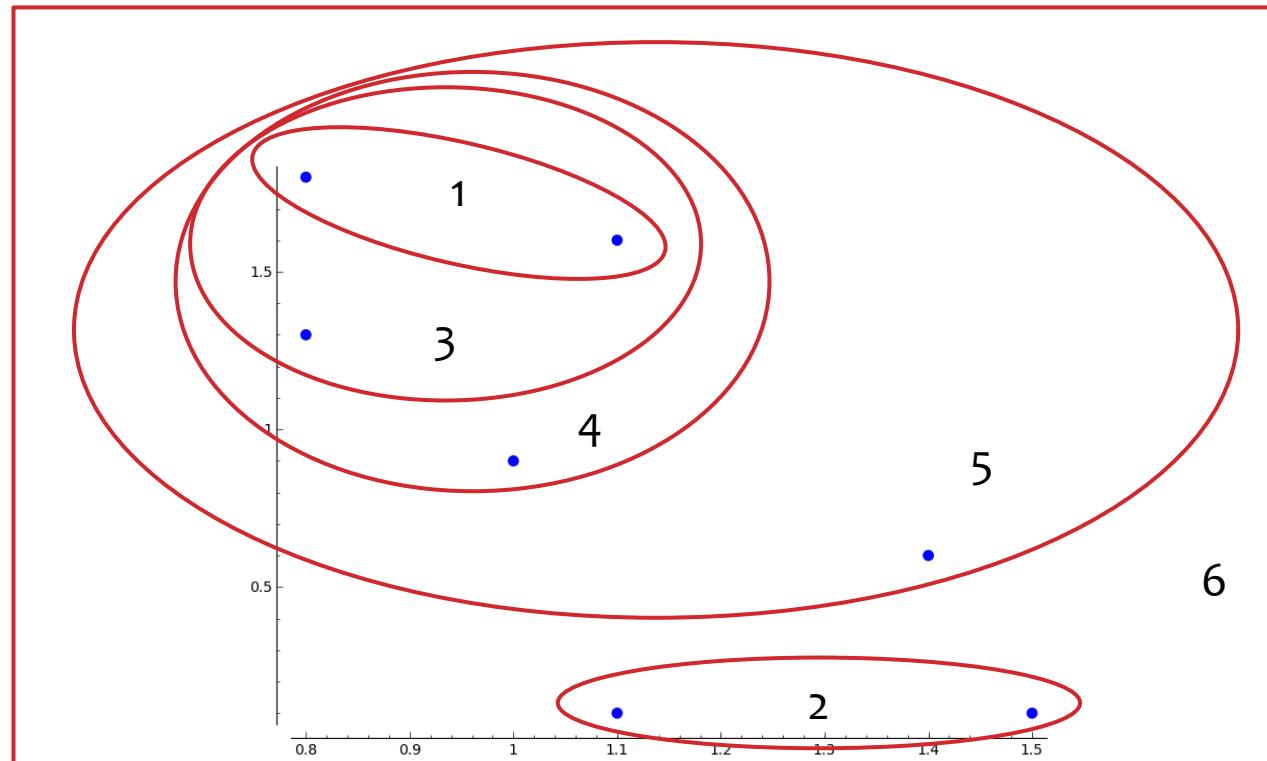


# Ejemplo

$[[0.8,1.8],[1.1,1.6],[0.8,1.3],[1.0,0.9],[1.4,0.6],[1.5,0.1],[1.1,0.1]]$

- Recalculamos distancias y fusionamos clusters

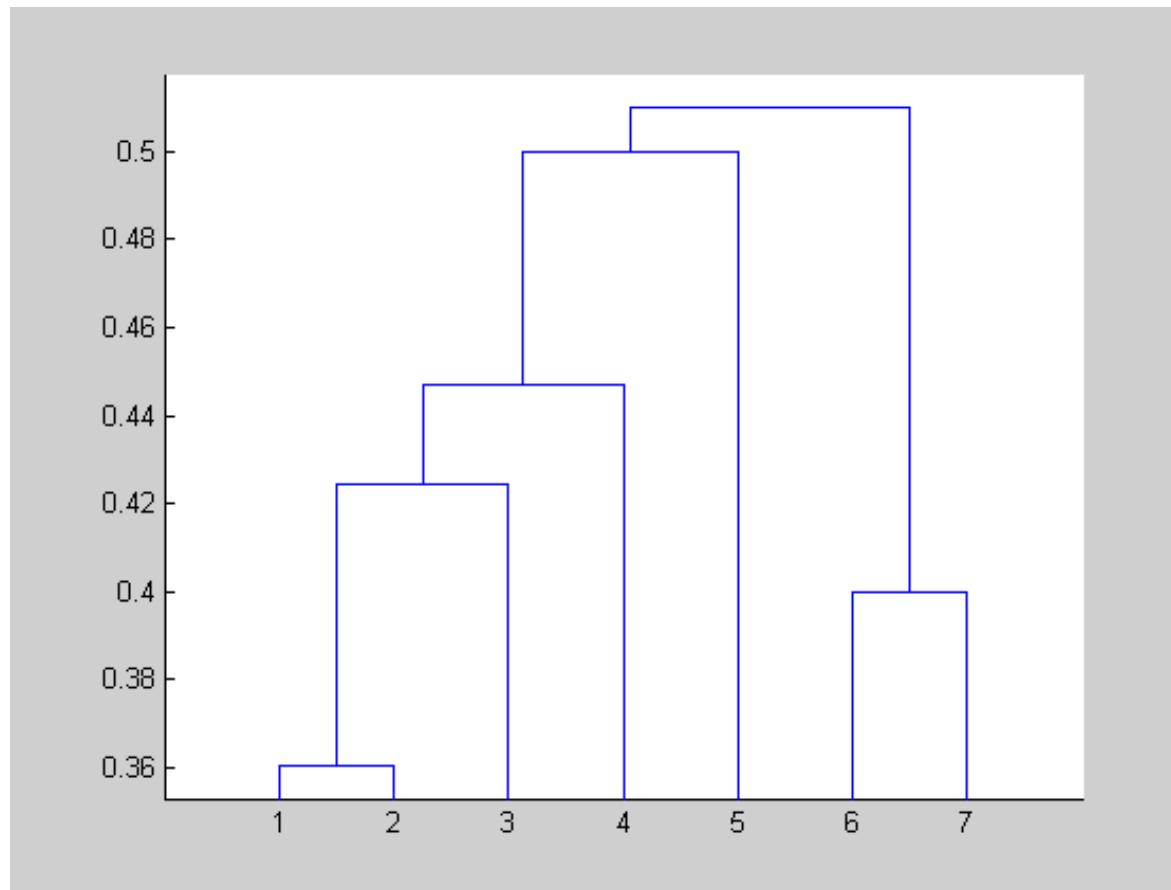
D5/D11		
D9	0.5831	
D5/D11		D9



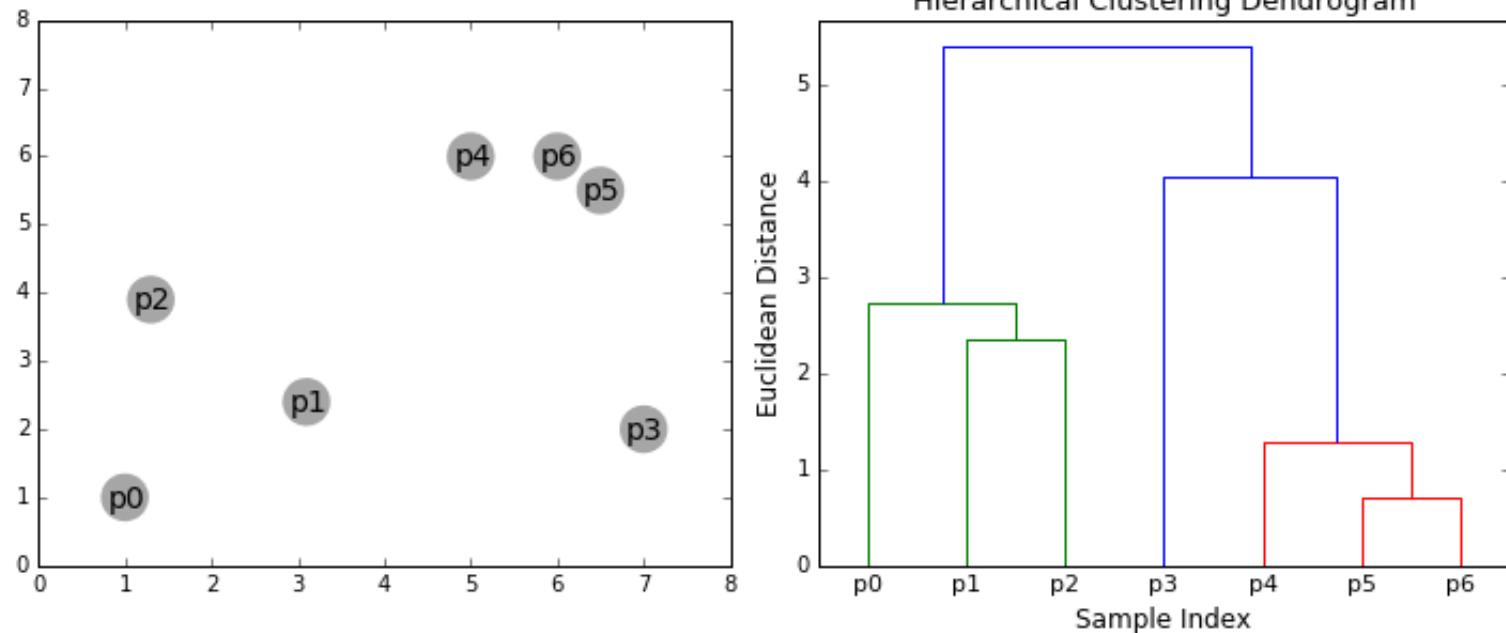
# Ejemplo

$[[0.8,1.8],[1.1,1.6],[0.8,1.3],[1.0,0.9],[1.4,0.6],[1.5,0.1],[1.1,0.1]]$

Calculamos dendrograma

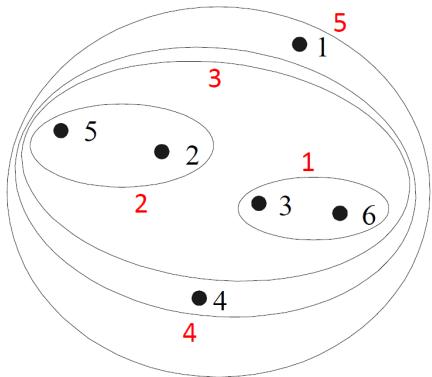


# Ejemplo

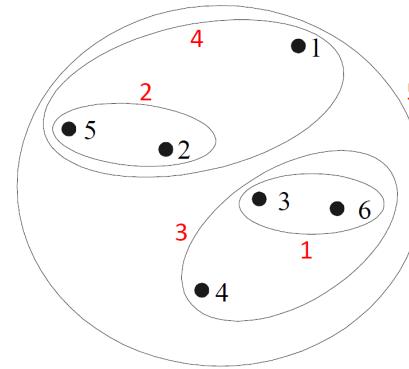


# Ejemplo

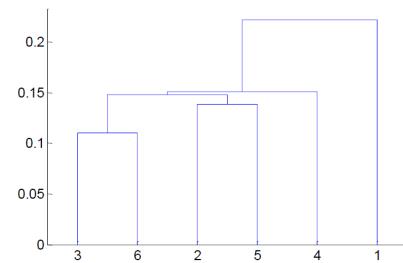
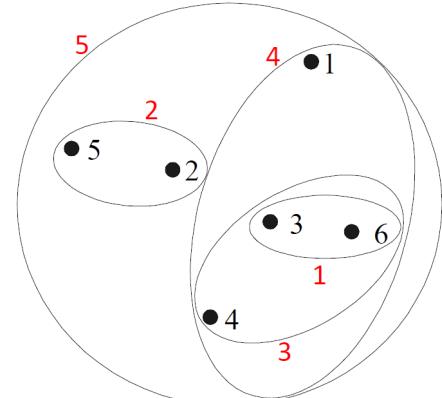
MIN



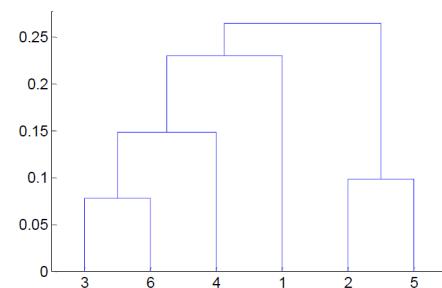
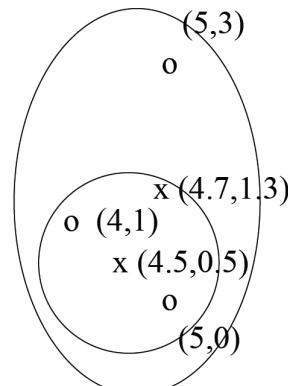
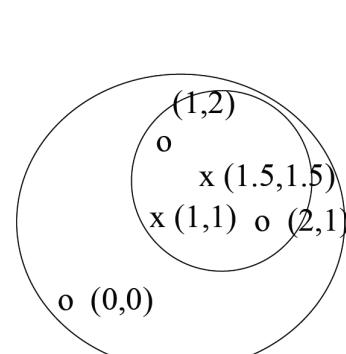
MAX



MEDIA



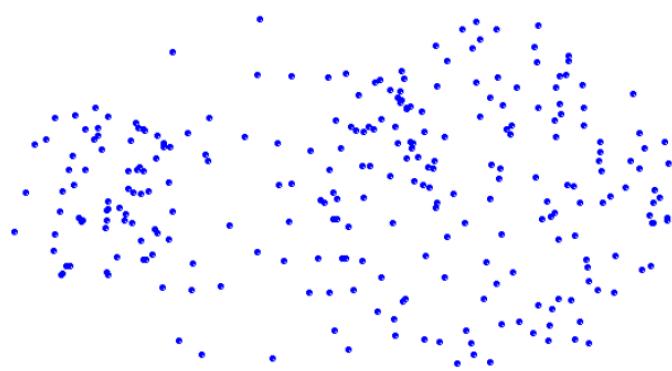
CENTROIDE  
Versión jerárquica de k-medias



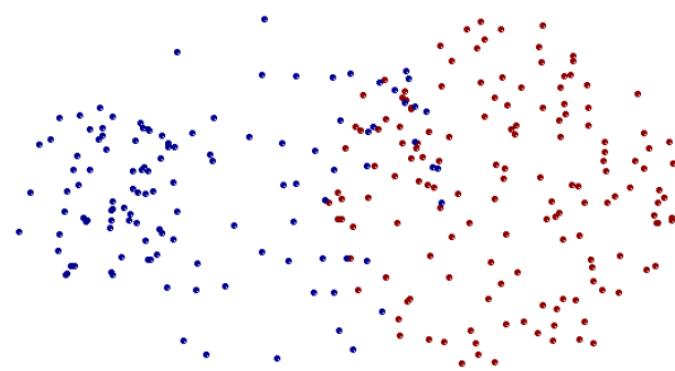
# MIN

## Limitaciones

Es **sensible al ruido y outliers**



Datos originales

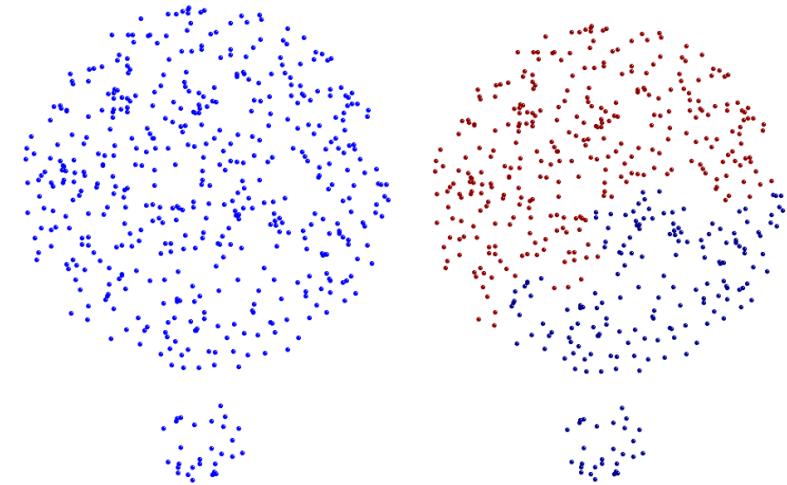


Dos clusters

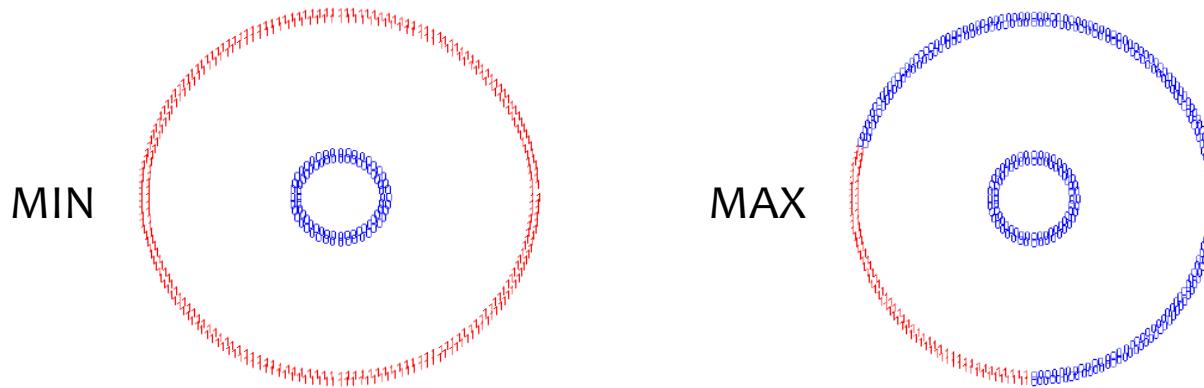
# MAX

## Limitaciones

Tiende a **romper clusters grandes**



Sesgado hacia **clusters convexos**



# Eficiencia

Principal inconveniente del clustering jerárquico:

## Baja escalabilidad

- $O(n^3)$  en tiempo
- $O(n^2 \log n)$  en tiempo y  $O(n^2)$  es espacio

Por este motivo, si se usa un método jerárquico para estimar el número de clusters (para utilizar k-means, por ejemplo), se suele emplear una **muestra de los datos** y no el conjunto de datos completo

# Otros algoritmos

Algunos métodos "escalables"

- **BIRCH:** Balanced Iterative Reducing and Clustering using Hierarchies (Zhang, Ramakrishnan & Livny, SIGMOD'1996)
- **ROCK:** RObust Clustering using linKs (Guha, Rastogi & Shim, ICDE'1999)
- **CURE:** Clustering Using REpresentatives (Guha, Rastogi & Shim, SIGMOD'1998)
- **CHAMELEON:** Hierarchical Clustering Using Dynamic Modeling (Karypis, Han & Kumar, 1999)

Trabajo evaluable (3 personas, 30 minutos). Algoritmos de clustering jerárquico. Considera los algoritmos anteriores y realiza un trabajo explicando su funcionamiento y presentando algunos ejemplos

# Algoritmo básico divisivo

**Algorithm** *GenericTopDownClustering*(Data:  $\mathcal{D}$ , Flat Algorithm:  $\mathcal{A}$ )  
**begin**

    Initialize tree  $\mathcal{T}$  to root containing  $\mathcal{D}$ ;

**repeat**

        Select a leaf node  $L$  in  $\mathcal{T}$  based on pre-defined criterion;

        Use algorithm  $\mathcal{A}$  to split  $L$  into  $L_1 \dots L_k$ ;

        Add  $L_1 \dots L_k$  as children of  $L$  in  $\mathcal{T}$ ;

**until** termination criterion;

**end**

Hay que elegir un método particional para escindir cada nodo  
Por ejemplo, 2-medias

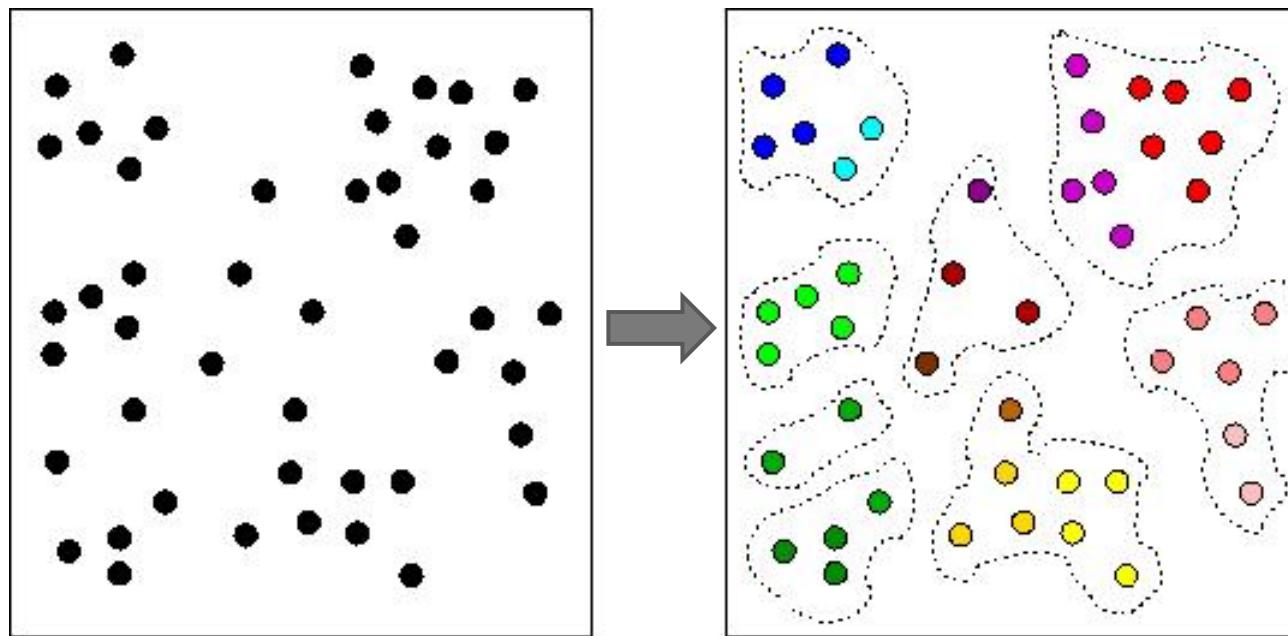
# Índice

- ❑ Concepto de clustering
- ❑ Distancias
- ❑ Clustering jerárquico
- ❑ **Clustering basado en representantes**
- ❑ Clustering basado en densidad
- ❑ Grid-based methods
- ❑ Clustering basado en modelos
- ❑ Evaluación/validación del clustering

# Clustering particional

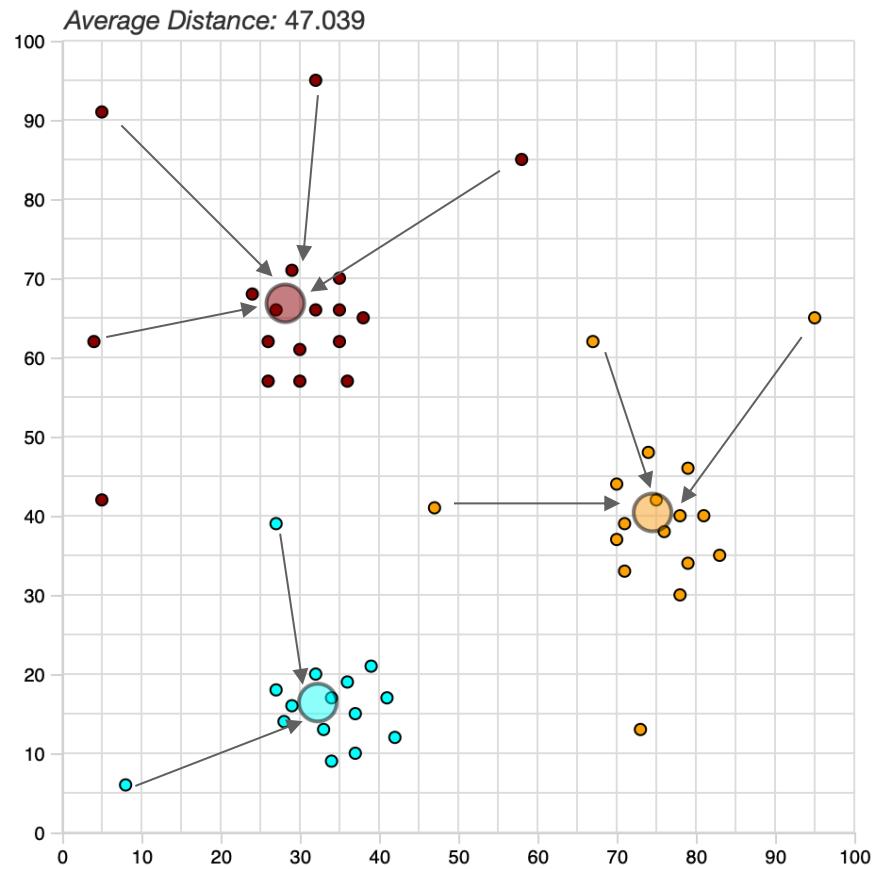
El objetivo del clustering/agrupamiento particional es dividir el conjunto de datos en un conjunto de clusters

$$\mathcal{C} = \{C_1, C_2, \dots, C_k\}$$



# Representative-based clustering

Están basados en la elección de representantes para cada cluster y la asignación de los datos a cada cluster dependiendo de la similitud/distancia a los representantes



# Representative-based clustering

El usuario debe seleccionar a priori:

- número de clusters/representantes
- los representantes iniciales

Y lo que se busca es iterar sobre esa elección hasta calcular unos representantes que minimicen

$$\sum_{i=1}^k \sum_{c \in C_i} d(c, r_i)$$

donde  $C_i$  es el cluster formado por los objetos cuyo representante más cercano, para una distancia  $d$ , es  $r_i$

# Representative-based clustering

**Algorithm** *GenericRepresentative*(Database:  $\mathcal{D}$ , Number of Representatives:  $k$ )

**begin**

    Initialize representative set  $S$ ;

**repeat**

        Create clusters ( $\mathcal{C}_1 \dots \mathcal{C}_k$ ) by assigning each point in  $\mathcal{D}$  to closest representative in  $S$  using the distance function  $Dist(\cdot, \cdot)$ ;

        Recreate set  $S$  by determining one representative  $\overline{Y}_j$  for each  $\mathcal{C}_j$  that minimizes  $\sum_{\overline{X}_i \in \mathcal{C}_j} Dist(\overline{X}_i, \overline{Y}_j)$ ;

**until** convergence;

**return** ( $\mathcal{C}_1 \dots \mathcal{C}_k$ );

**end**

**Criterio de parada del bucle:**  
mejora de la función objetivo por debajo de un umbral

# Representative-based clustering

- ❑ El paso más costoso es la asignación a los clusters
    - Calculo distancia a representantes de cada objeto
  - ❑ Suelen terminar en pocas iteraciones
- 
- ❑ Eficiencia en  $O(nkdi)$  donde:
    - $n$  es el número de ejemplos
    - $i$  es el número de iteraciones
    - $d$  es el número de atributos
    - $k$  es el número de clusters

# K-medias

- Algoritmo clásico de clustering, los representantes son los centros (geométricos) de los clusters
- El **centro (centroide)** de un cluster tiene como coordenadas la media aritmética de las coordenadas de sus elementos

$$\text{Centro}(x_1, x_2, \dots, x_k) = \frac{x_1 + x_2 + \dots + x_k}{k}$$

- Se utiliza con atributos reales y con la distancia Euclidea
- Se busca minimizar el error, SSE = Sum of Square Errors

$$\text{SSE} = \sum_{i=1}^k \sum_{a \in C_i} d_{\text{EUCLIDEA}}(a, \text{centro}[i])^2$$

# K-medias

---

**Algorithm** Algoritmo  $k$ -medias

**Input:**  $C = \{c_i\}_{i=1,\dots,m}$  ejemplos,  $k$  entero,  $d$  distancia

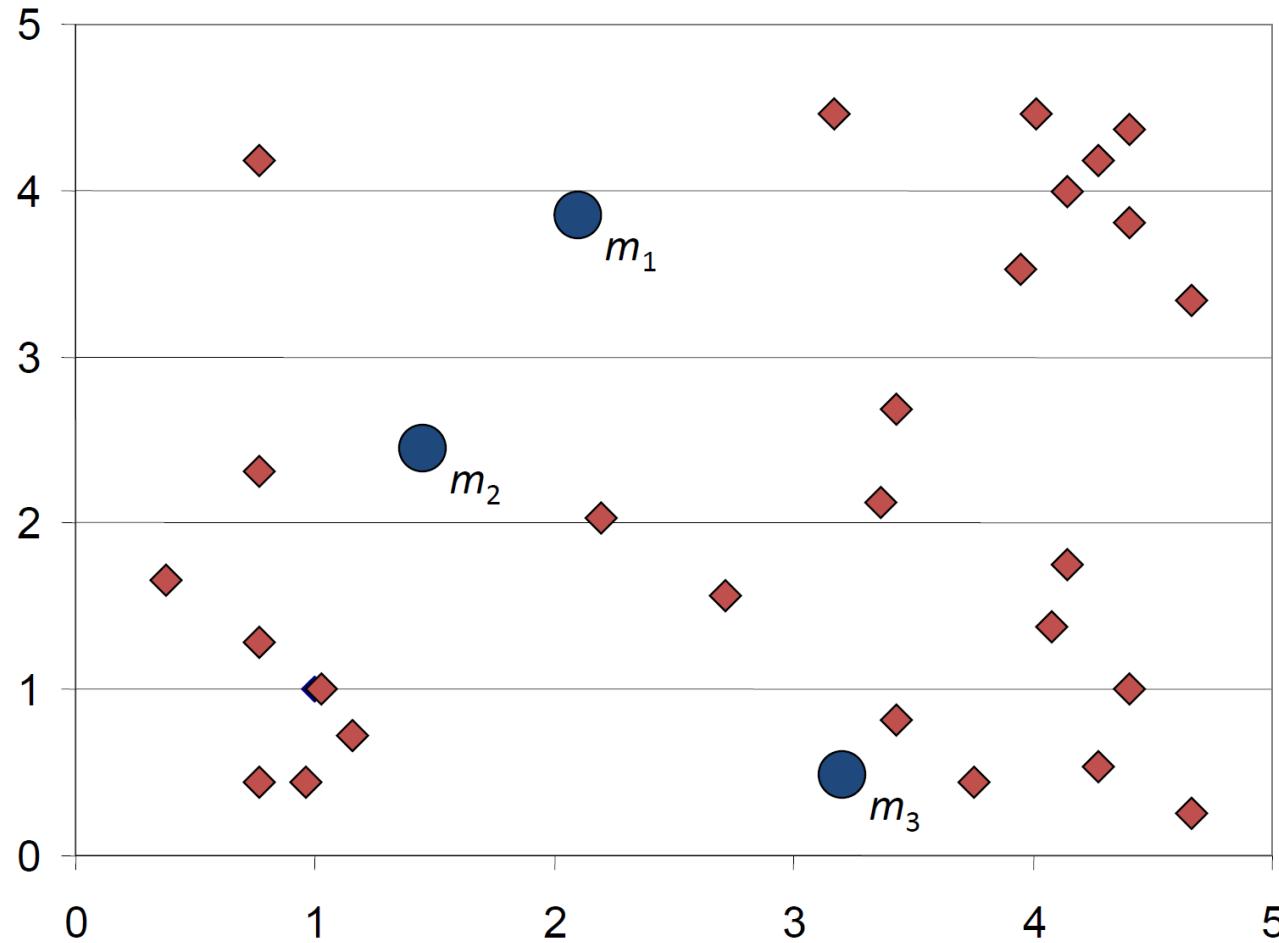
**Output:**  $C_1, C_2, \dots, C_k$  clusters

```
1: for  $1 \leq i \leq k$  do
2:   Seleccionar  $r_i$ , no necesariamente en  $C$ 
3:   centro[ $i$ ]  $\leftarrow r_i$ 
4: while Cambien los clusters  $C_1, C_2, \dots, C_k$  do
5:   for  $1 \leq i \leq k$  do
6:     //los cluster son los elementos mÁs cercanos a cada centro
7:      $C_k \leftarrow \{c \in C \text{ tales que } d(c, \text{centro}[k]) \leq d(c, \text{centro}[j]) \quad \forall j \neq k\}$ 
8:   for  $1 \leq i \leq k$  do
9:     //se recalculan los centros
10:    centro[ $i$ ]  $\leftarrow$  punto medio de  $C_k$ 
11: return  $C_1, \dots, C_k$ 
```

---

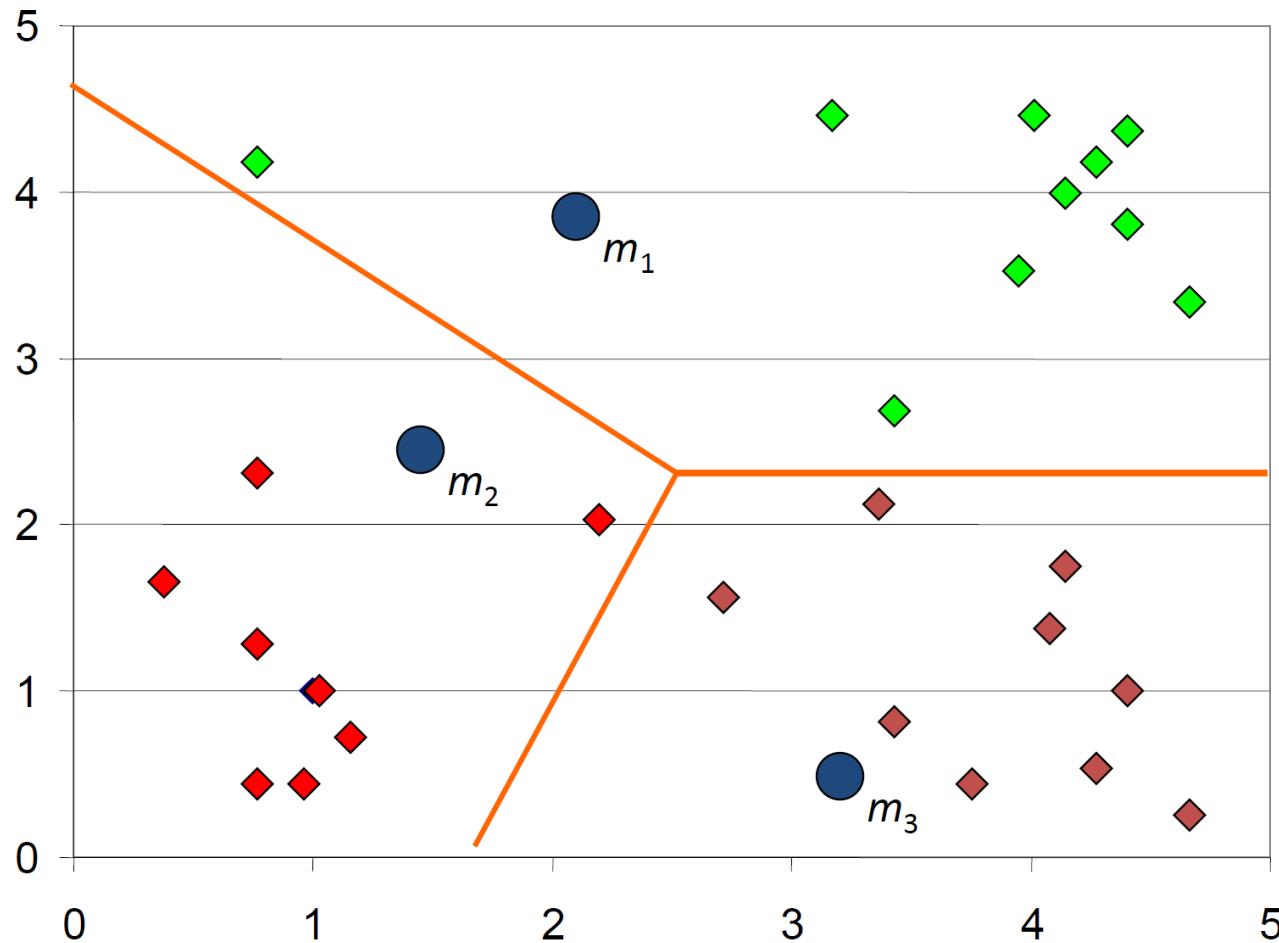
# K-medias

## Inicialización – Selección de centroides



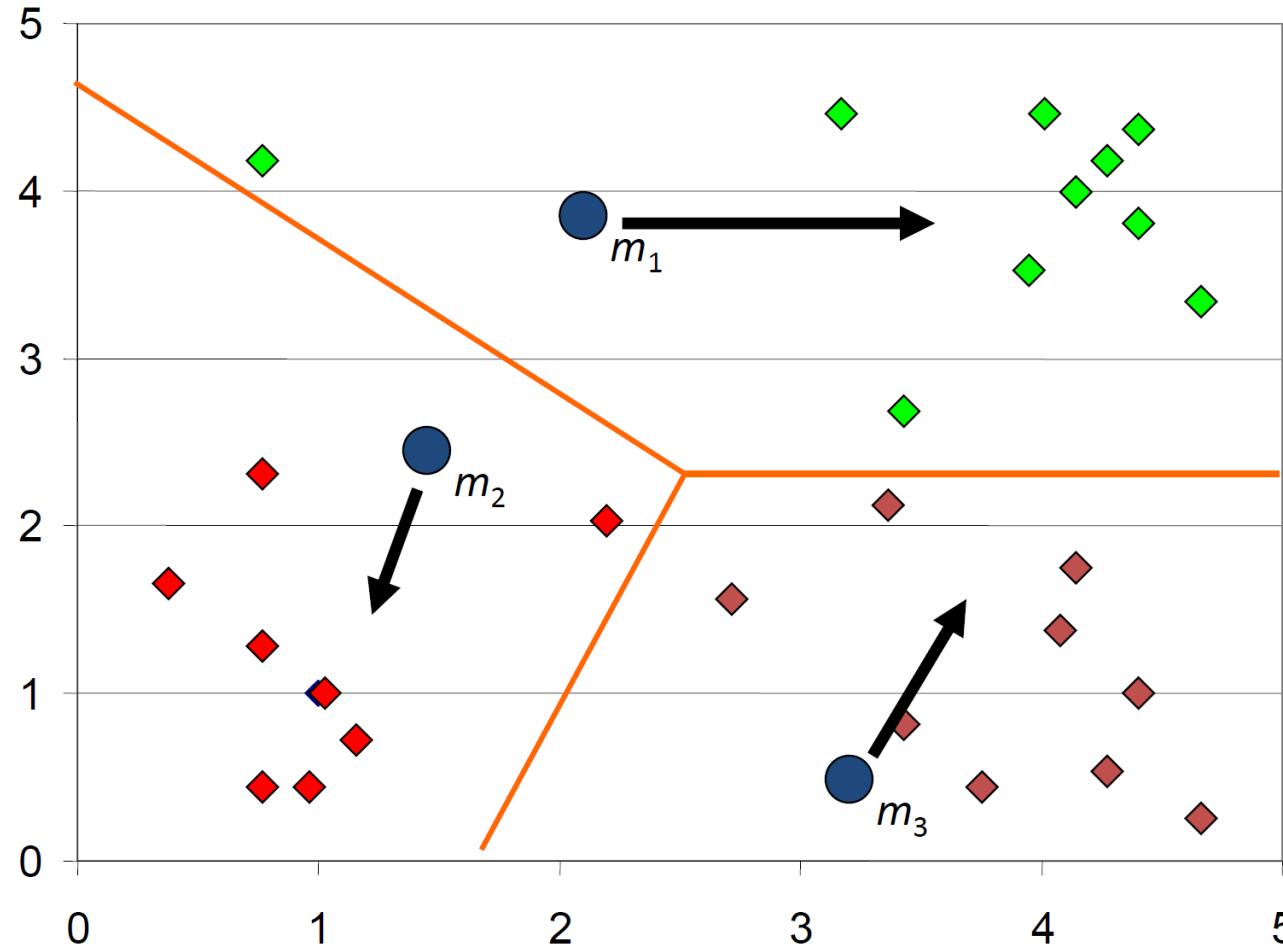
# K-medias

**Bucle** – Establecer clusters (distancia euclídea)



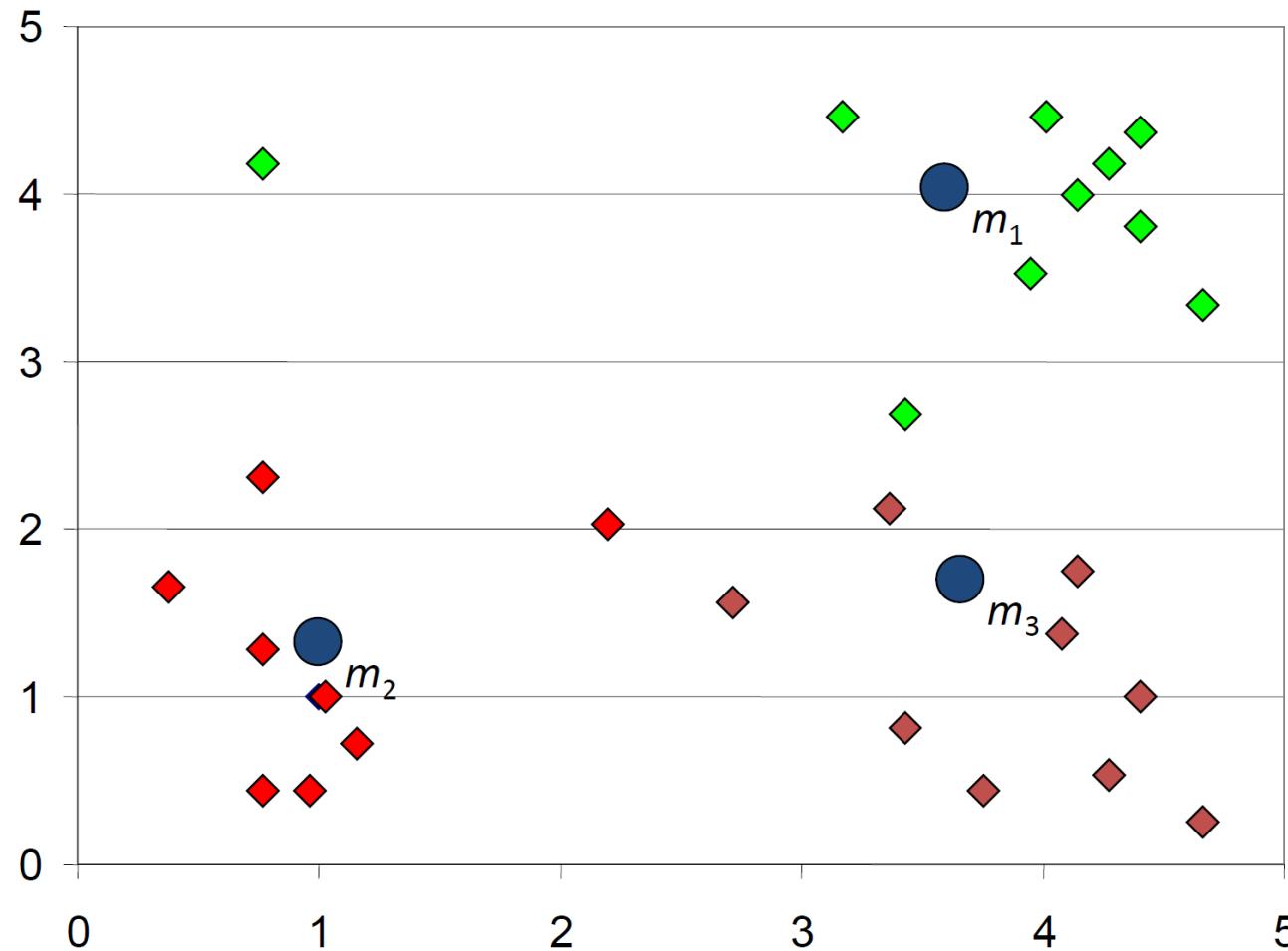
# K-medias

## Bucle – Actualizar centroides



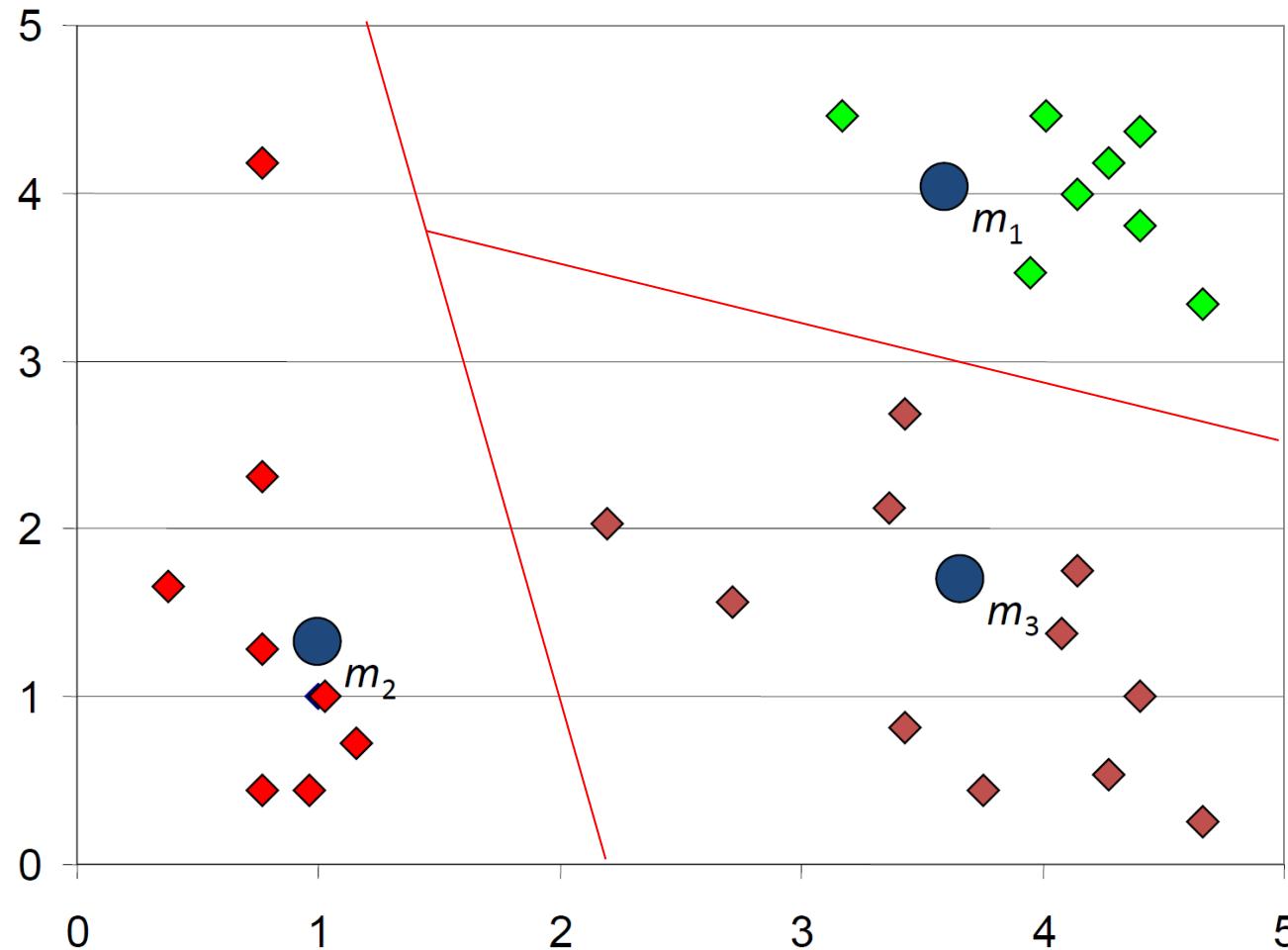
# K-medias

Bucle – Actualizar centroides



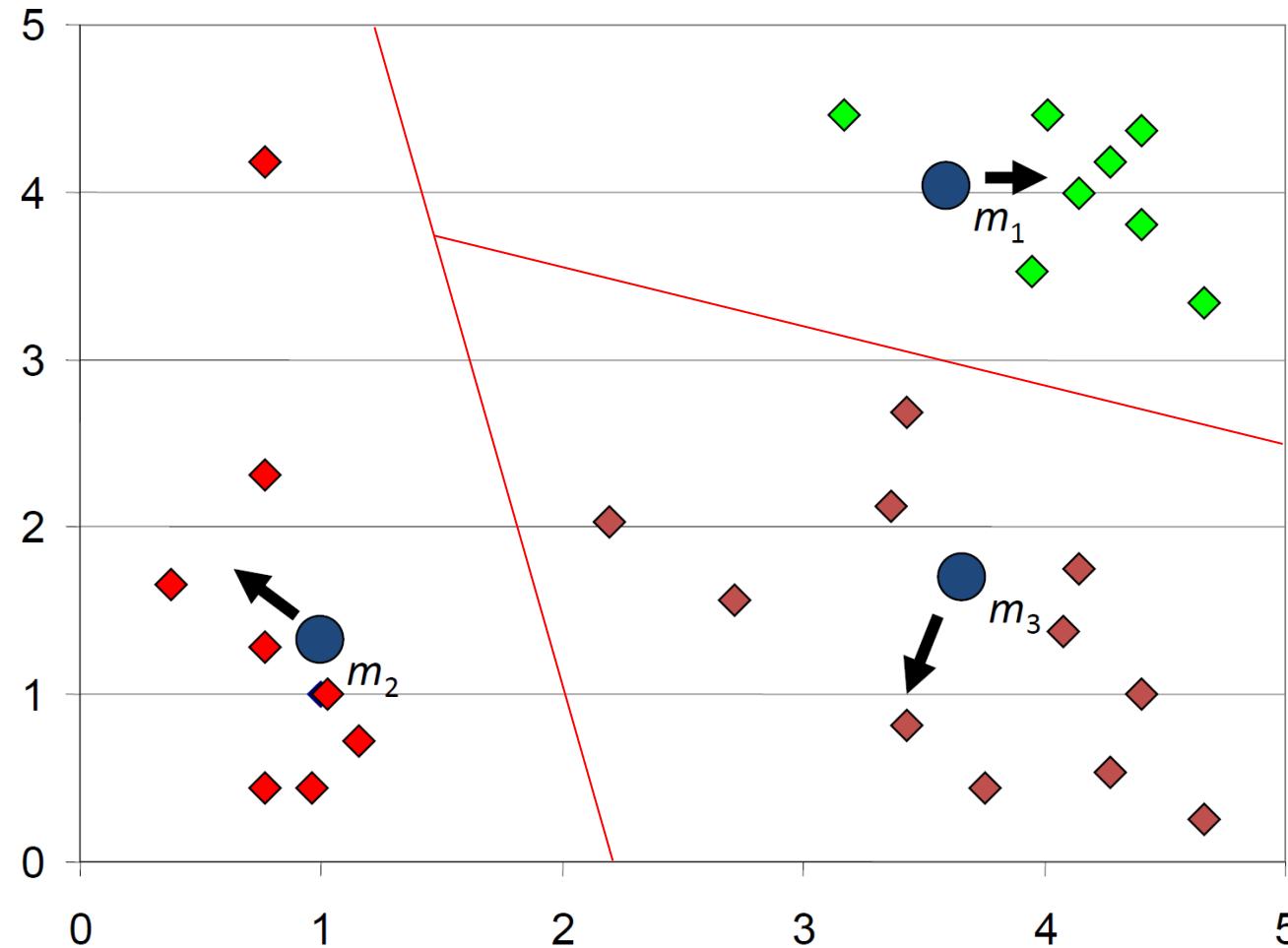
# K-medias

Bucle – Establecer clusters



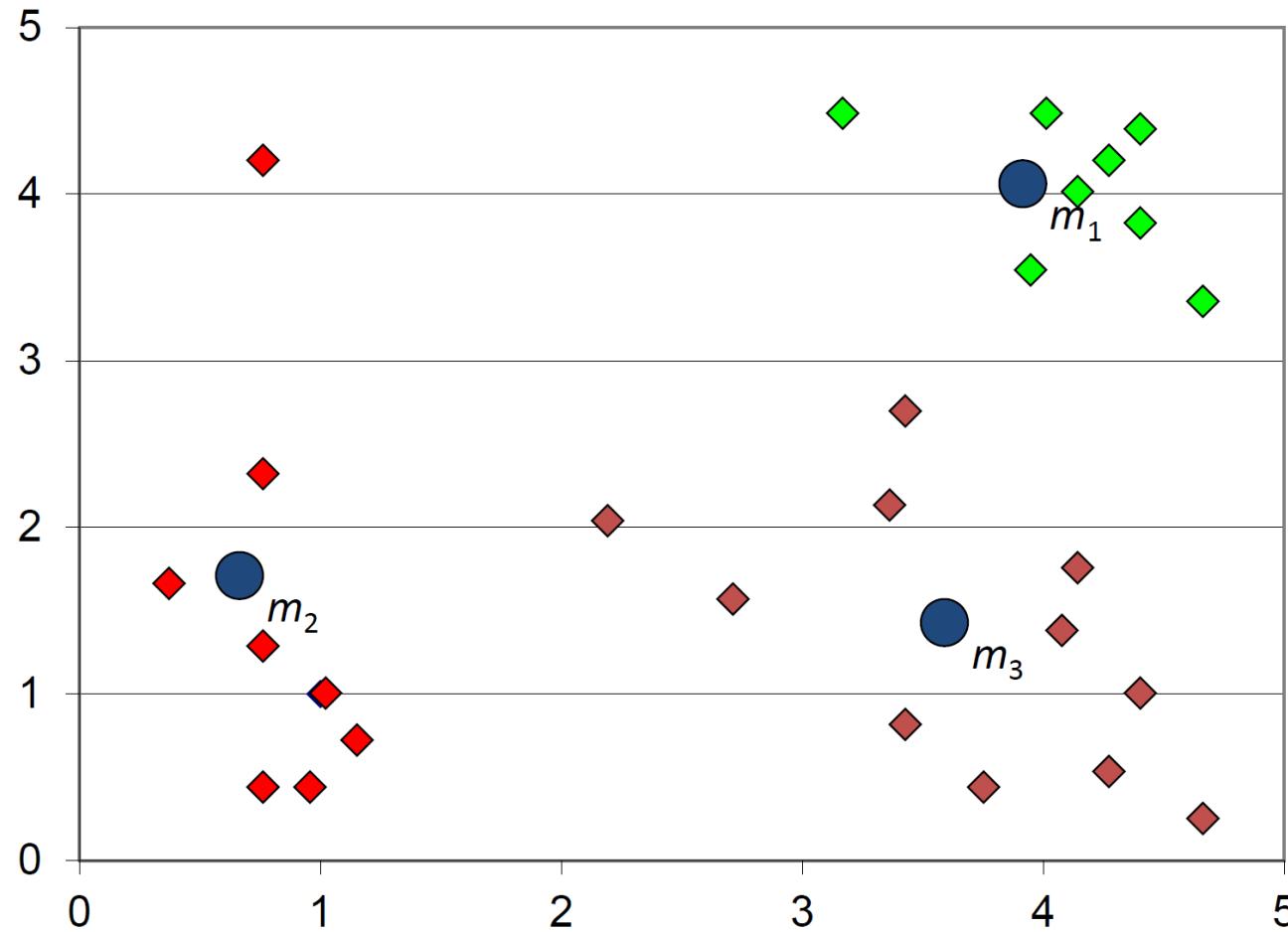
# K-medias

## Bucle – Actualizar centroides



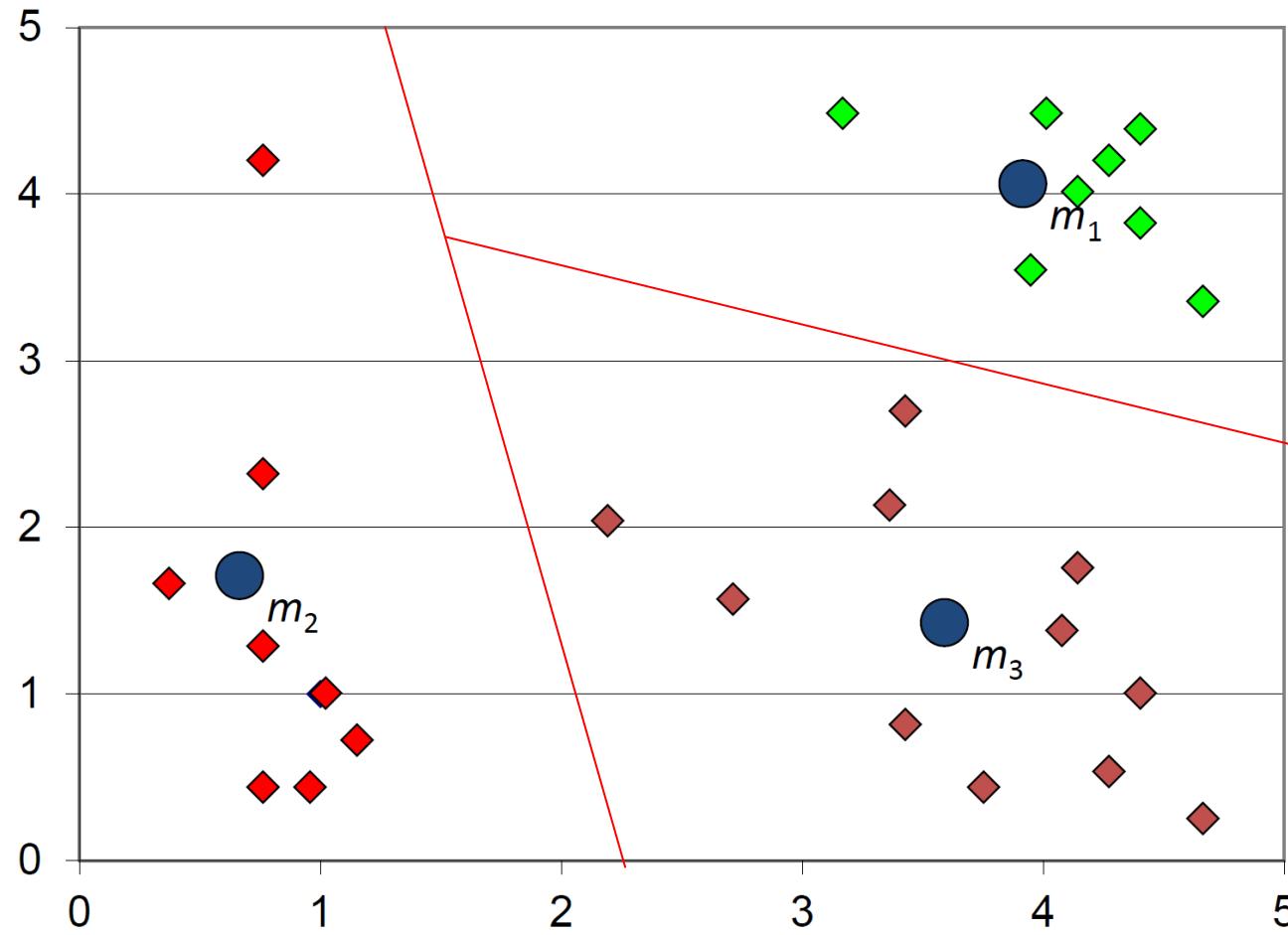
# K-medias

## Bucle – Actualizar centroides

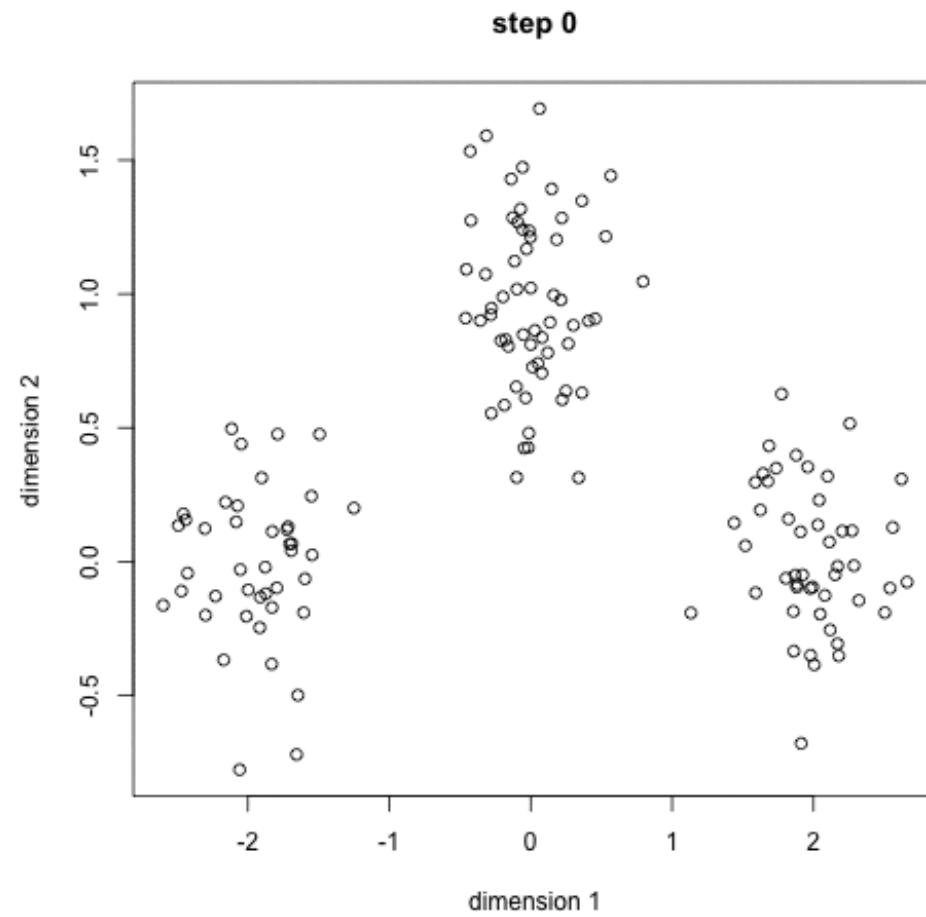


# K-medias

**Bucle** – Actualizar clusters (no cambian = fin)

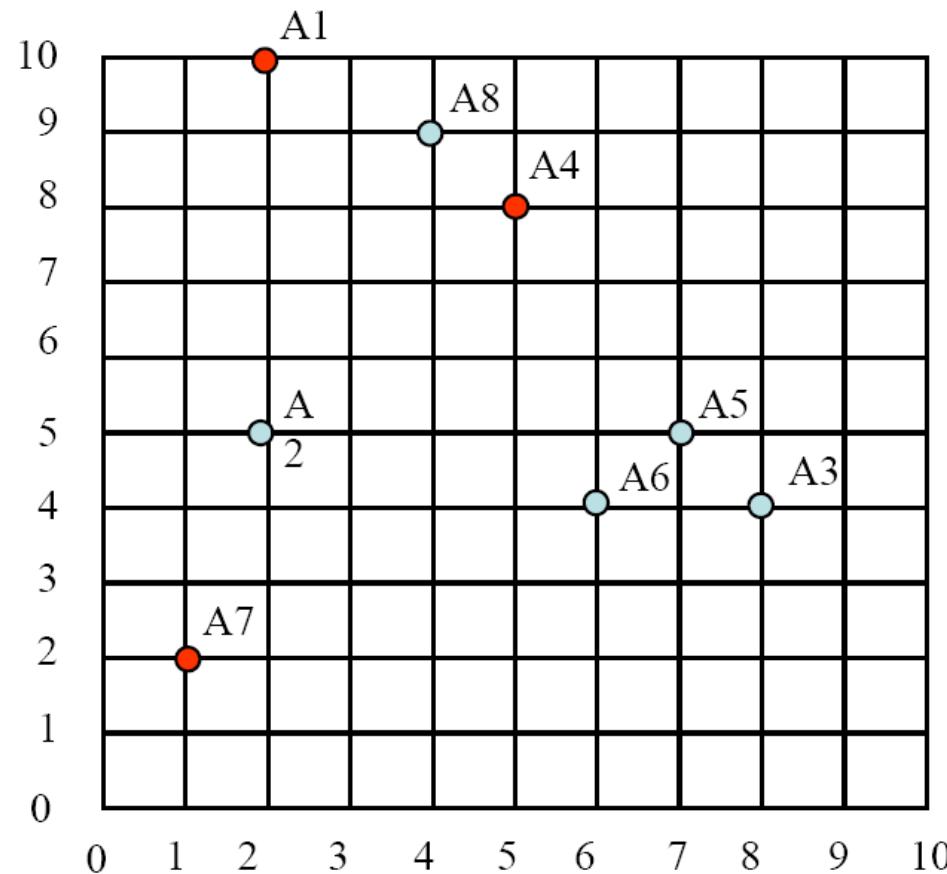


# K-medias



# K-medias

**Ejercicio.** Agrupar los datos usando el algoritmo k-medias.  
Los centros iniciales son los puntos marcados en rojo. Usar la distancia Euclídea

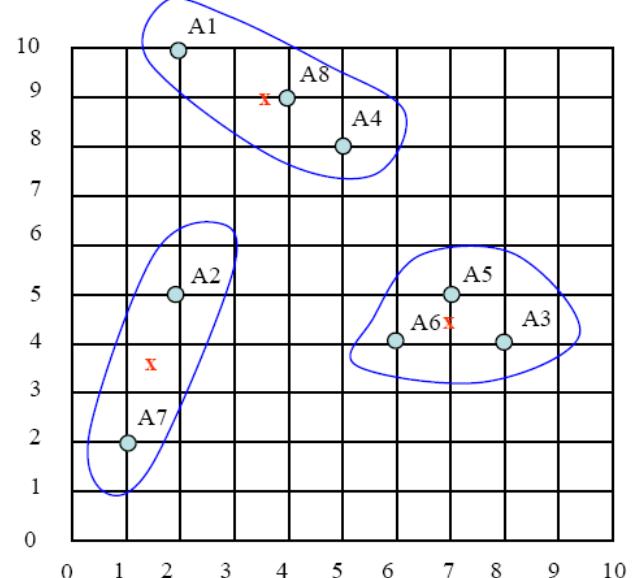
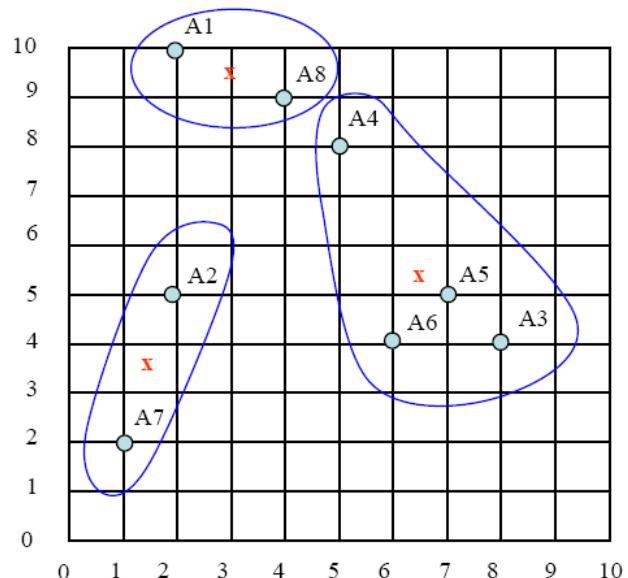
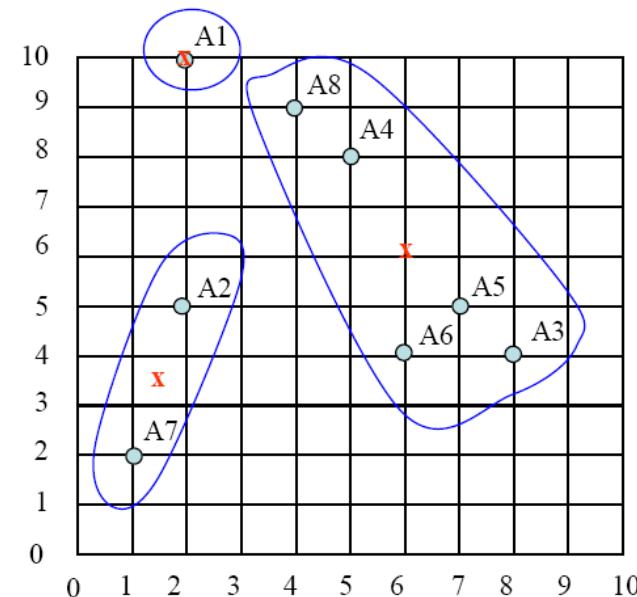
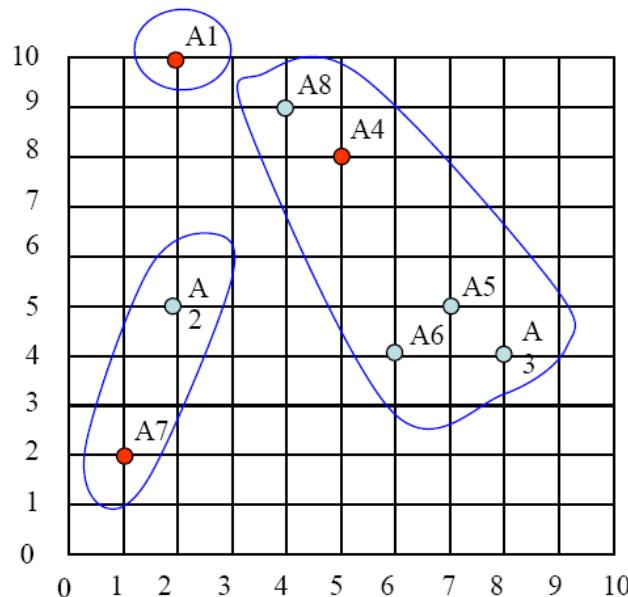


# K-medias

Con la distancia Euclídea

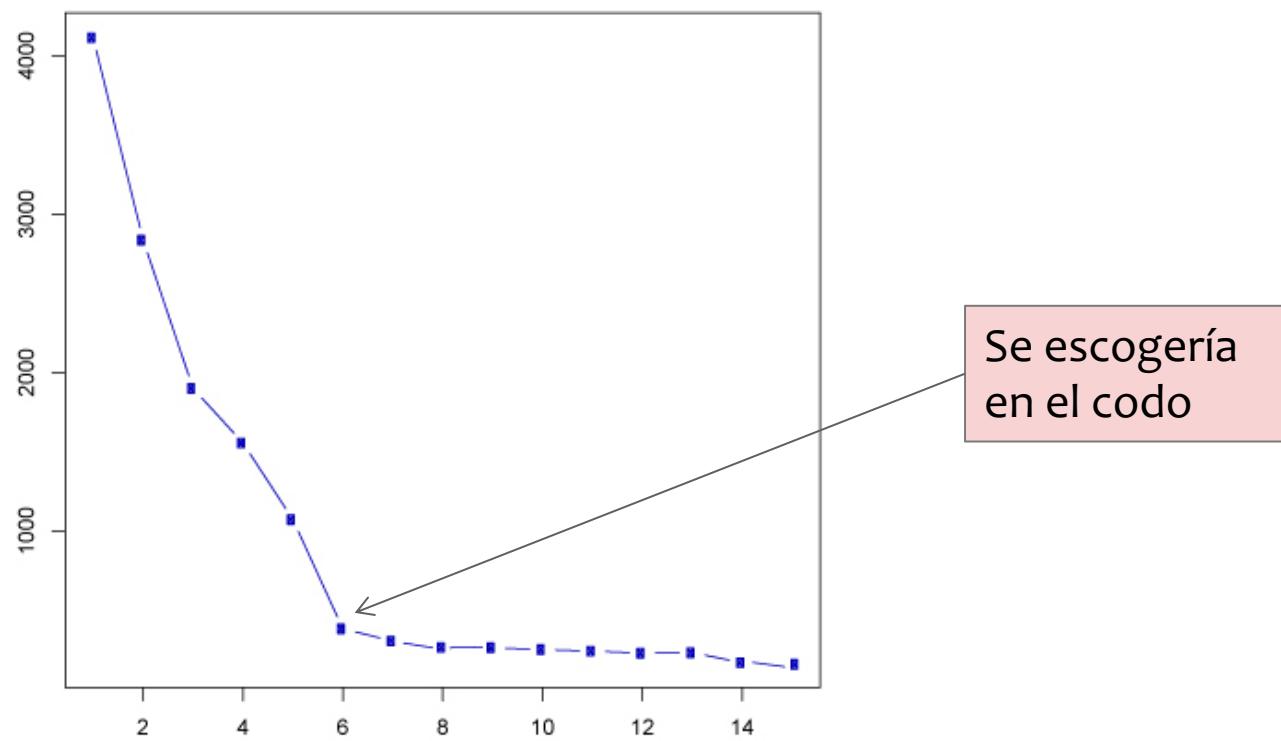
	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

# K-medias



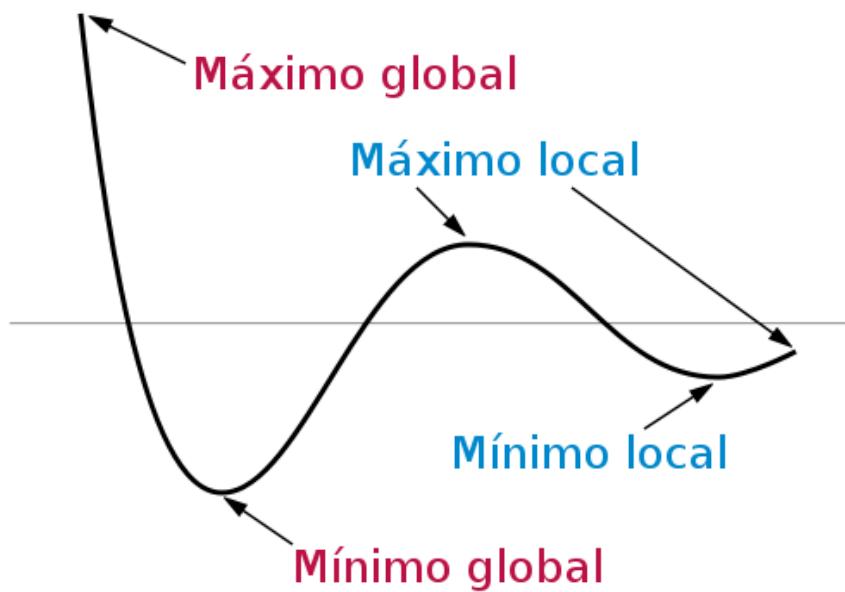
# K-medias

- Problema: hay que especificar el **número de clusters** al principio
  - Es difícil saberlo a priori
  - Solución: calcular el **error** para varios k y se escoge a partir del cual no disminuye de forma significativa



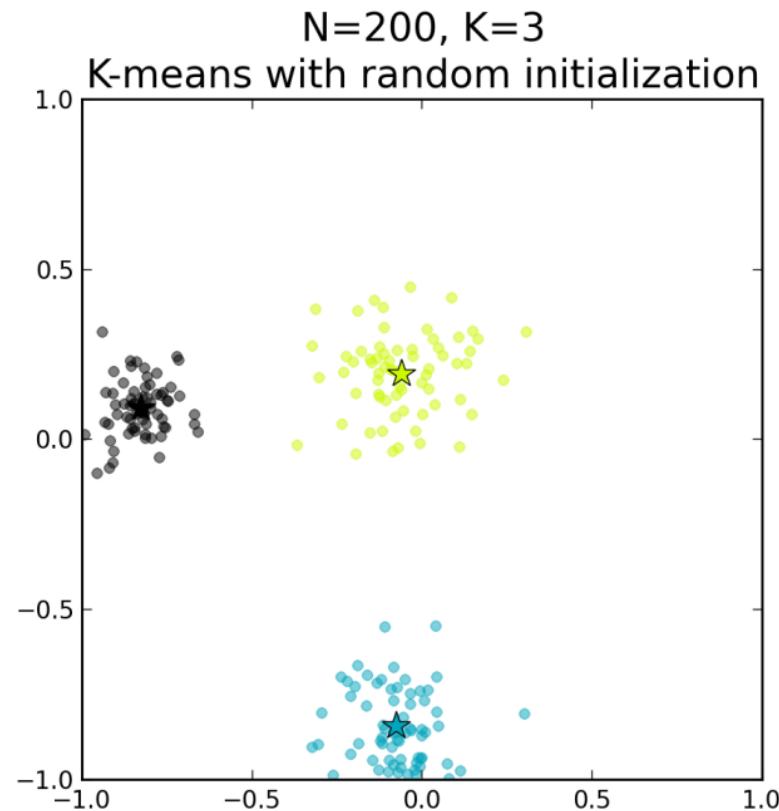
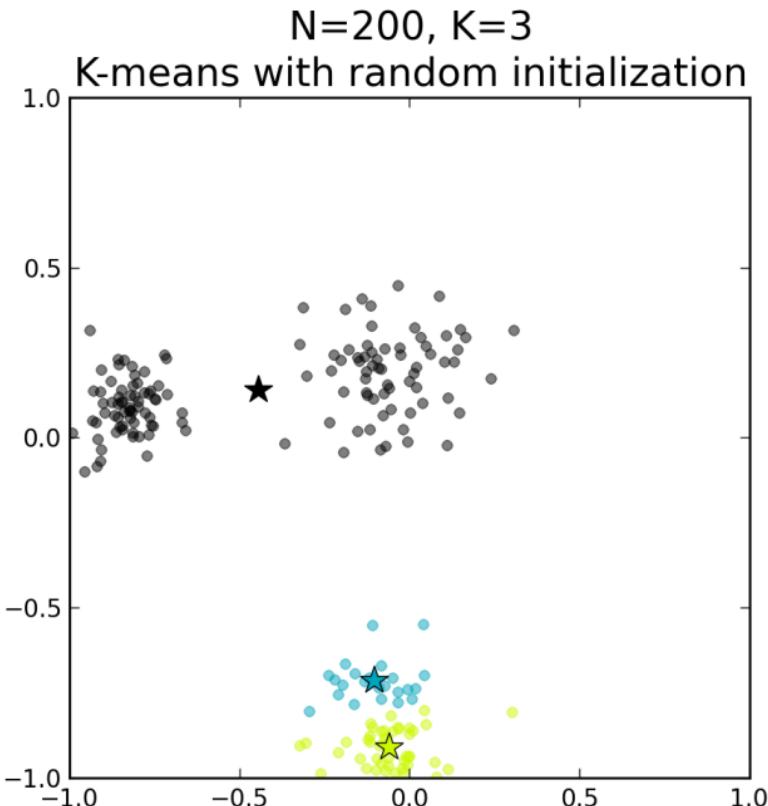
# K-medias

- Problema: hay que especificar el **número de clusters** al principio
  - Es difícil saberlo a priori
  - Solución: calcular el **error** para varios  $k$  y se escoge a partir del cual no disminuye de forma significativa
  - Pero... puede **no converger al mínimo global** de la función error. Puede ser un **mínimo local**



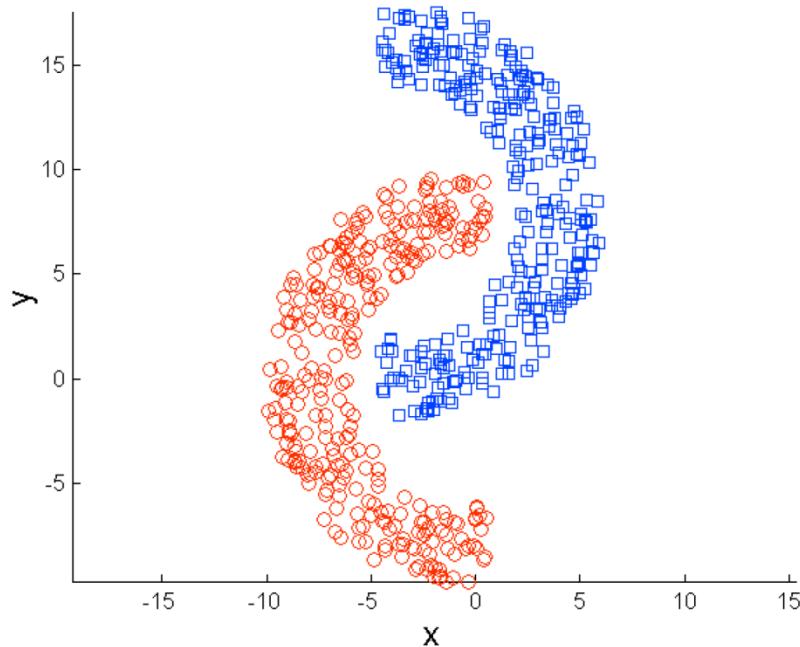
# K-medias

- ❑ Otro problema: el resultado **depende** fuertemente de los centroides iniciales

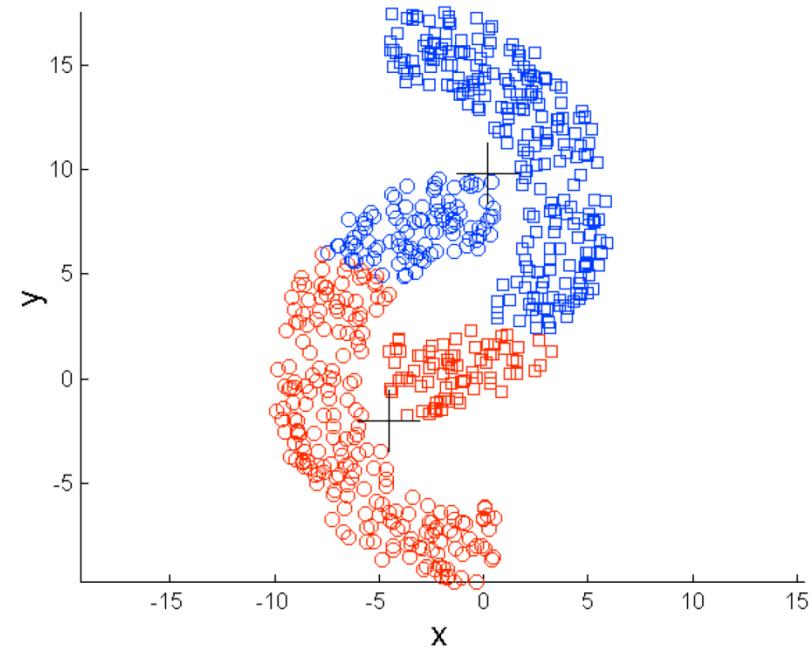


# K-medias

- ❑ Más problemas: no es adecuado para **detectar clusters no convexos, o de formas irregulares**



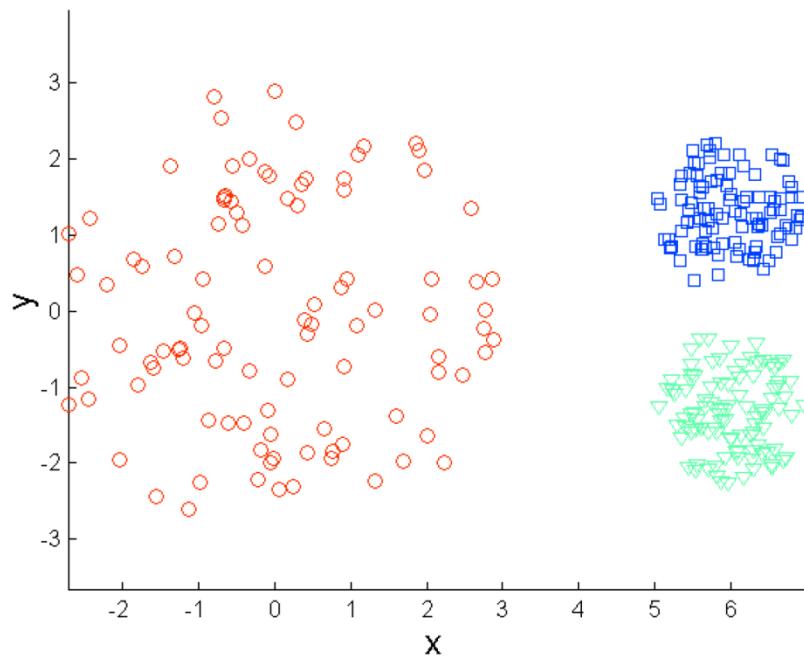
Puntos originales



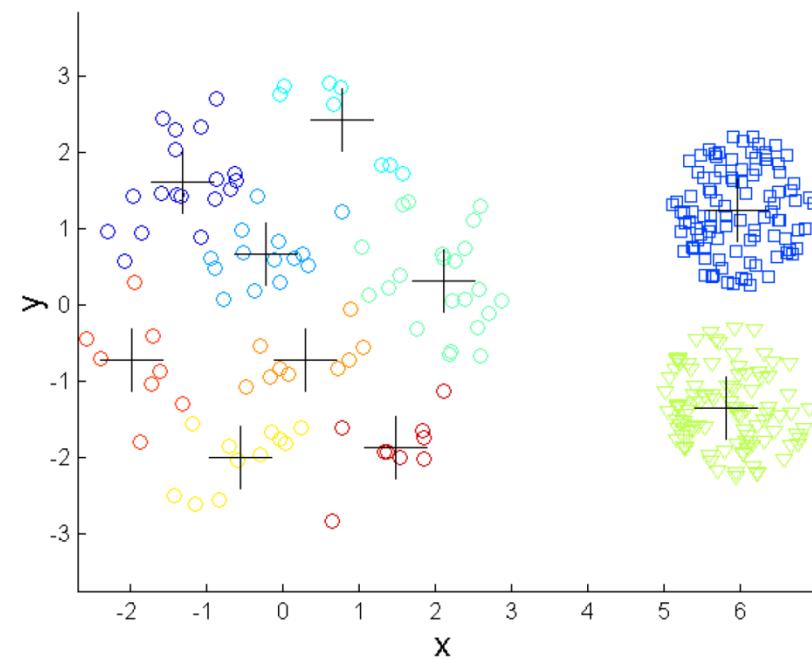
K-medias (2 clusters)

# K-medias

- ❑ Más problemas: no es adecuado para detectar clusters de diferente densidad



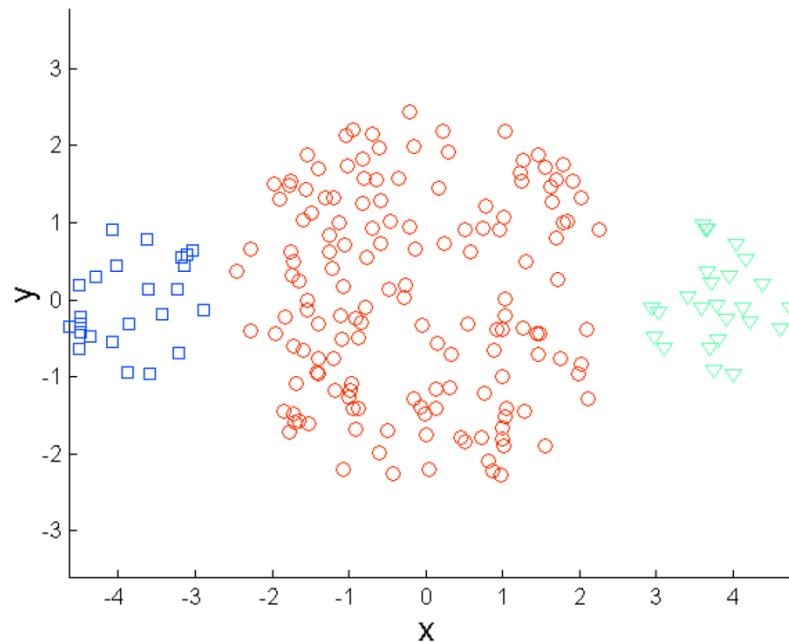
Puntos originales



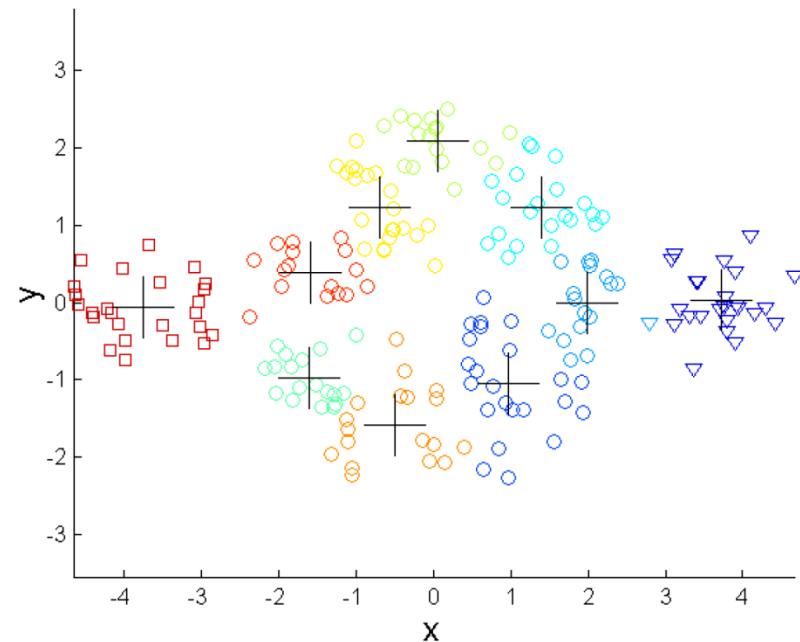
K-medias

# K-medias

- ❑ Más problemas: no es adecuado para detectar clusters de diferente tamaño



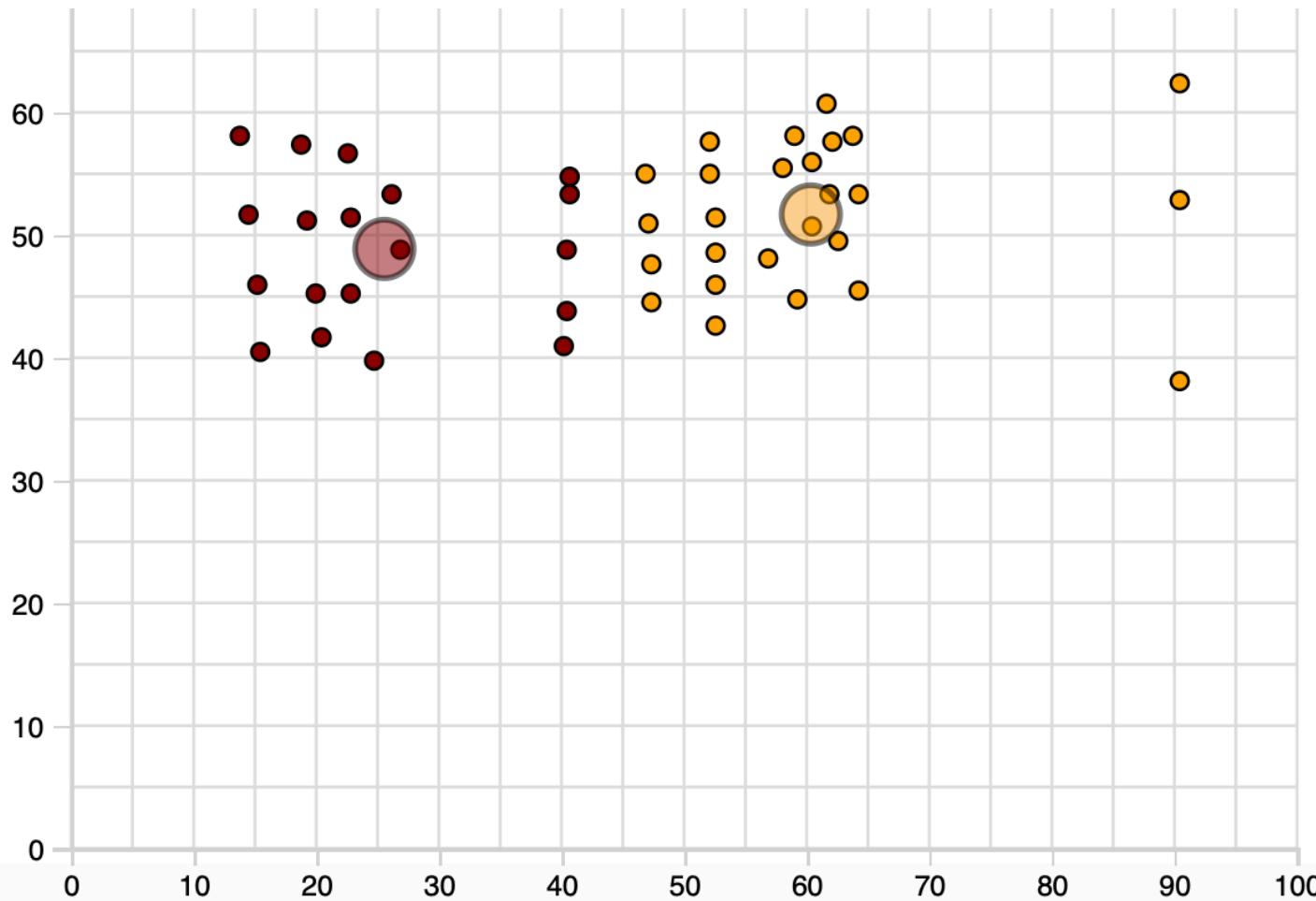
Puntos originales



K-medias

# K-medias

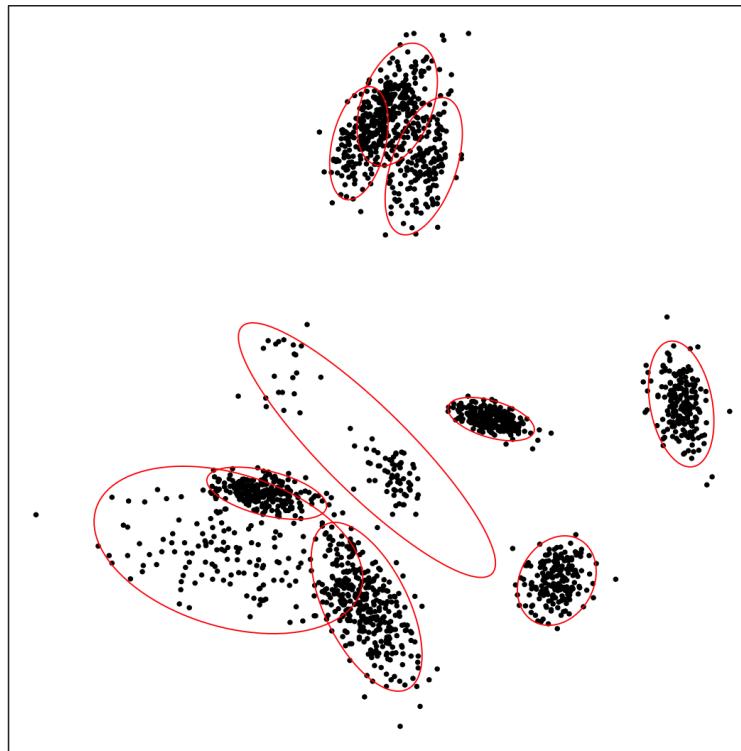
- ❑ Más problemas: sensible a outliers



# Mahalanobis k-means

Si utilizamos la distancia de Mahalanobis, se pueden detectar clusters con forma elipsoidal

[10.1016/j.spl.2013.09.026](https://doi.org/10.1016/j.spl.2013.09.026)



Incluso puede funcionar bien con formas arbitrarias

# Kernel k-means

Podemos generalizarlo para detectar cualquier forma mediante el uso de *kernels* (*kernel trick*)

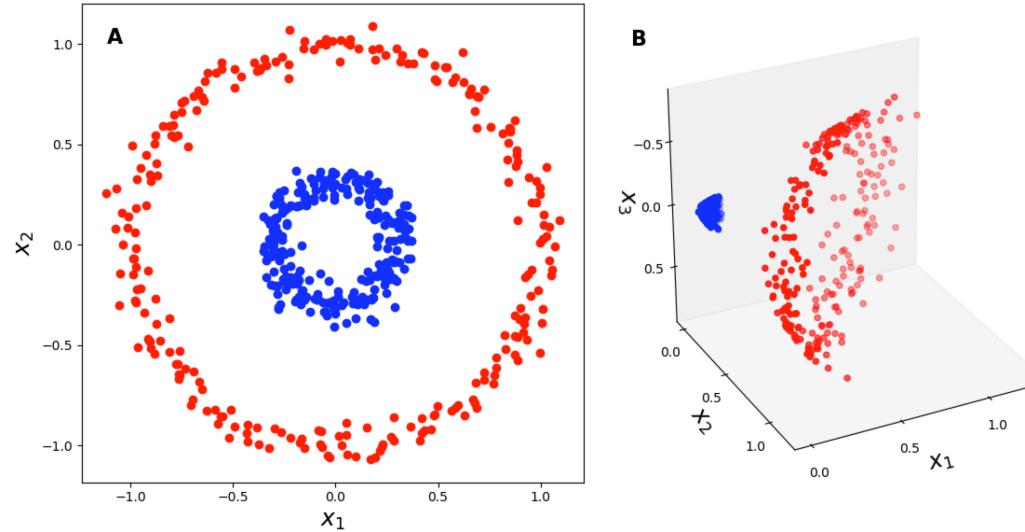
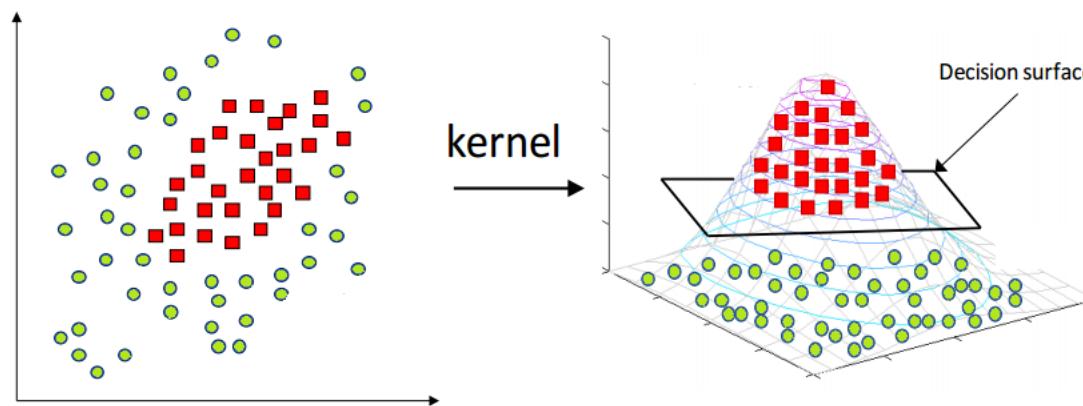
Matemáticamente, un **kernel** no es más que un embebimiento entre espacios métricos

$$\Psi : (A, \langle \cdot, \cdot \rangle_A) \longrightarrow (B, \langle \cdot, \cdot \rangle_B)$$

Es decir,  $\langle x, y \rangle_A = \langle \Psi(x), \Psi(y) \rangle_B$  para todo  $x, y \in A$

Transformamos los datos de un espacio a otro más sencillo para realizar el clustering

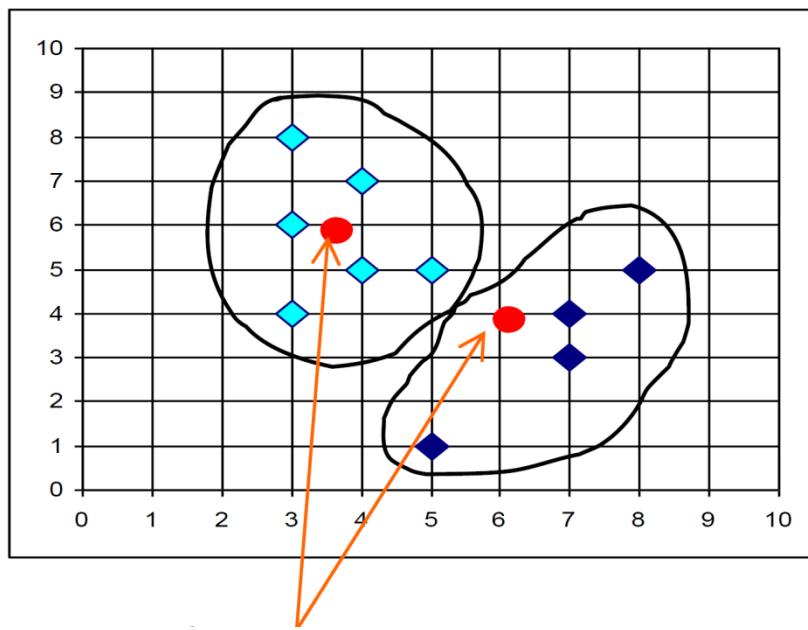
# Kernel k-means



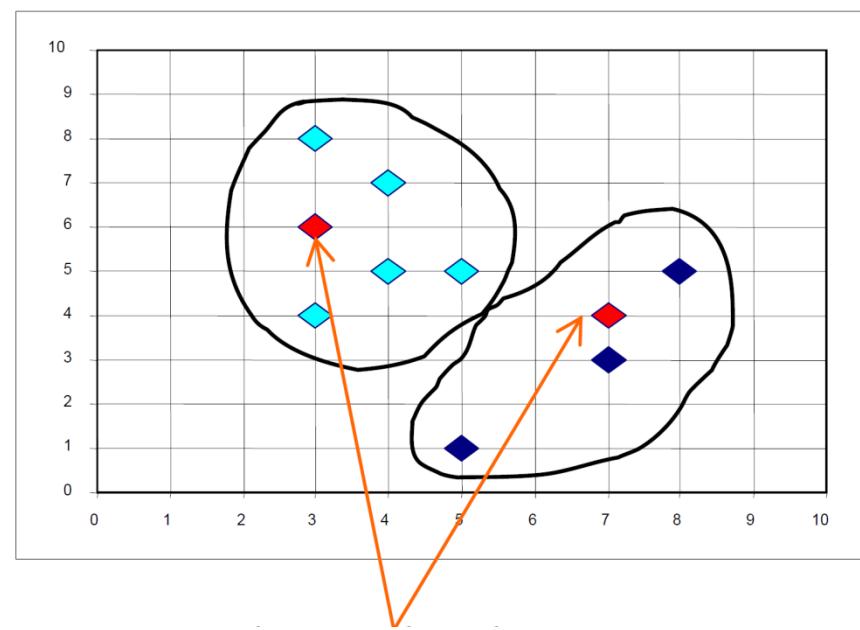
La dificultad está en encontrar la transformación adecuada

# K-medoids

- Variación de k-medias para relajar la influencia de outliers
- En vez de calcular la distancia al centro del cluster, se seleccionan k-elementos del conjunto de datos (medoids) que harán el papel de centros



k-medoids



k-medoids

# K-medoids

- ❑ Los menoides se cambiarán por otros elementos si se reduce el error de la nueva configuración

$$\text{Error} = \sum_{i=1}^k \sum_{a \in C_i} dist(a, \text{medoid}[i])$$

$$\text{Coste}(\text{medoid}[i] \leftarrow x_i) = \text{Error}(\text{medoid}[i] \leftarrow x_i) - \text{Error}$$

- ❑ Esto tiene un alto coste computacional (NP-Hard) por lo que se suele dar como dato una matriz de distancias

# K-medoids (PAM, Partitioning Around Medoids)

---

**Algorithm 1** Algoritmo  $k$ -medoids

---

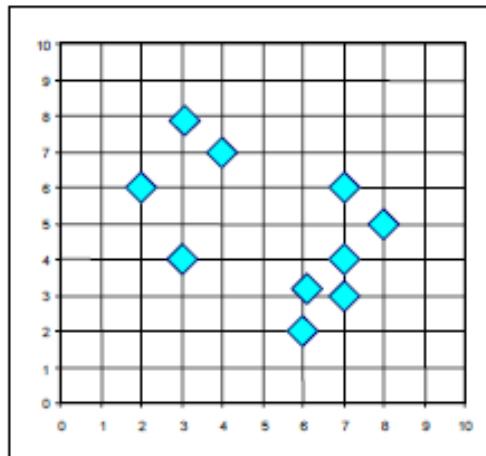
**Input:**  $C = \{c_i\}_{i=1,\dots,m}$  ejemplos,  $k$  entero,  $d$  distancia

**Output:**  $C_1, C_2, \dots, C_k$  clusters

```
1: for  $1 \leq i \leq k$  do
2:   Seleccionar  $c_i \in C$ 
3:   medoid[i]  $\leftarrow c_i$ 
4: for  $1 \leq i \leq k$  do
5:    $C_k \leftarrow \{c \in C \text{ tales que } d(c, \text{medoid}[k]) \leq d(c, \text{medoid}[j]) \quad \forall j \neq k\}$ 
6: while Cambien los clusters  $C_1, C_2, \dots, C_k$  do
7:   for  $1 \leq i \leq k$  do
8:     for Todo elemento  $x$  que no es medoid do
9:       Calcular el coste,  $S$ , de cambiar medoid[i] por  $x$ 
10:      if  $S < 0$  then
11:        medoid[i]  $\leftarrow x$ 
12: for  $1 \leq i \leq k$  do
13:   //recalculamos los clusters
14:    $C_k \leftarrow \{c \in C \text{ tales que } d(c, \text{medoid}[k]) \leq d(c, \text{medoid}[j]) \quad \forall j \neq k\}$ 
15: return  $C_1, \dots, C_k$ 
```

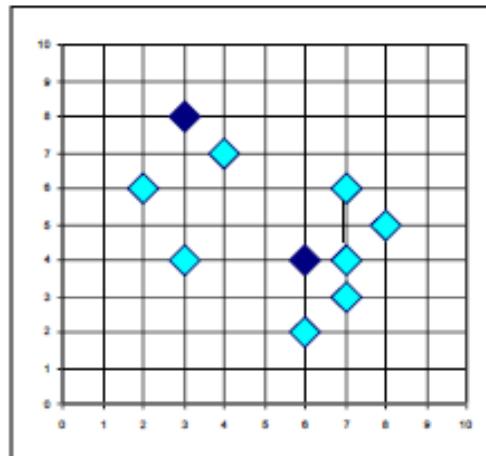
Ascenso de  
colinas

# K-medoids



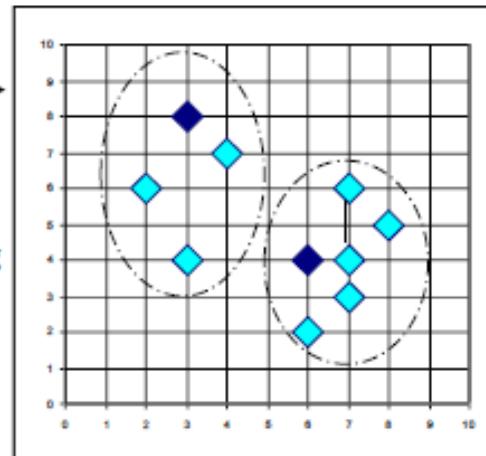
K=2

Arbitrary  
choose k  
object as  
initial  
medoids



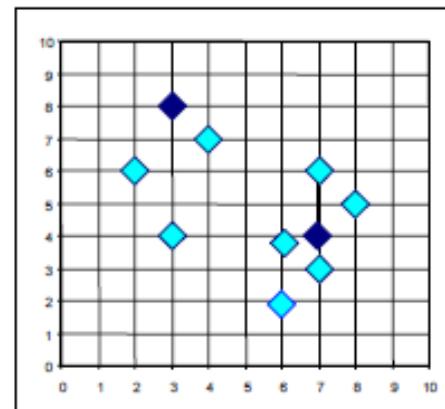
Total Cost = 26

Assign  
each  
remaining  
object to  
nearest  
medoids

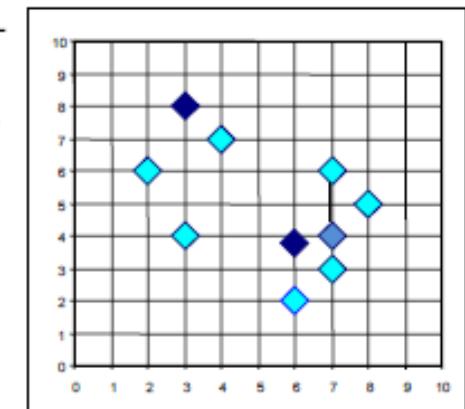


Total Cost = 20

Randomly select a  
nonmedoid object,  $O_{random}$



Compute  
total cost of  
swapping



Do loop

Until no change

Swapping  $O$   
and  $O_{random}$   
If quality is  
improved.

# CLARA (Kauffmann, Rousseauw, 1986)

CLARA (Clustering LARge Applications) es una modificación para mejorar la eficiencia de PAM

- Realizamos  $m$  muestreos ( $40+2k$  objetos) y aplicamos PAM a cada uno de ellos. Para cada conjunto de medoides, asignamos cada dato a su medoide más cercano.
- Obtenemos entonces  $m$  agrupamientos. Nos quedamos con el que tiene menor distancia media a medoides.
- A partir de este agrupamiento se genera un nuevo proceso de muestreo y una nueva iteración.

La complejidad de cada iteración está en  $O(kS^2 + k(n - k))$ , donde  $S$  es el tamaño de la muestra

La efectividad de CLARA depende del tamaño de la muestra, si bien es fácilmente escalable y puede trabajar con grandes BBDD

# CLARANS (Ng, Han, 2002)

CLARANS (Clustering Large Applications based upon RANdomized Search) intenta mejorar la selección de medoides (muestreo) de CLARA

Se considera la búsqueda de los medoides óptimos como un proceso de búsqueda en un árbol donde cada nodo es un conjunto de k medoides.

**Ejercicio evaluable (2 personas, 20 minutos). Clarans.**

Buscar la referencia

- R. Ng and J. Han, CLARANS: A method for clustering objects for spatial data mining, IEEE Transactions on Knowledge and Data Engineering 14 (5) 2002, 1003-1016.

Realizar las siguientes tareas: explicar la motivación de su diseño dada por los autores, explicación clara y detallada del algoritmo, explicación del ajuste de parámetros, inventar un pequeño ejemplo para mostrar su ejecución

# K-medianas

- ❑ Es una variante de k-medias, pero utilizando las medianas en vez del centro
- ❑ La mediana de un conjunto de datos es el valor que deja el 50% de los datos por encima
- ❑ Se utiliza para minimizar la desviación absoluta respecto al “centro”

$$\sum_{i=1}^k \sum_{a \in C_i} |a - \text{mediana}[i]|$$

- ❑ Conviene utilizarlo si la distancia es la distancia Manhattan

# K-modas

- Variante para tratar valores nominales
- Se considera la moda en vez de la media
- Es necesario establecer una distancia que trabaje con datos categóricos

age	income	student	credit_rating
< = 30	high	no	fair
< = 30	high	no	excellent
31..40	high	no	fair
> 40	medium	no	fair
> 40	low	yes	fair
> 40	low	yes	excellent
31..40	low	yes	excellent
< = 30	medium	no	fair
< = 30	low	yes	fair
> 40	medium	yes	fair
< = 30	medium	yes	excellent
31..40	medium	no	excellent
31..40	high	yes	fair

moda  
( $\leq 30$ , medium, yes, fair)

# ISODATA (Jacobs et al., 2000)

JOURNAL OF MAGNETIC RESONANCE IMAGING 11:425–437 (2000)

ISODATA es un método relacionado con k-means en el que se permite la escisión/unión de clusters

Paraméetros de entrada:

- Mínimo número de datos en un cluster
- Número de clusters deseados (no tiene que ser el final)
- Coeficiente para permitir escisión de un cluster
- Coeficiente para permitir unión de clusters
- Número máximo de clusters que pueden ser unidos
- Número máximo de iteraciones

# ISODATA (Jacobs et al., 2000)

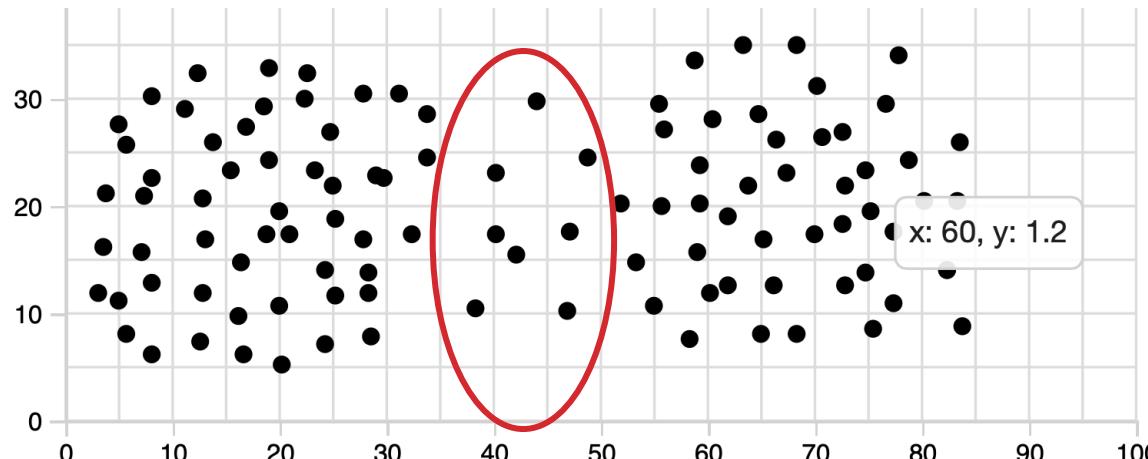
Pasos a seguir (genéricos):

1. Crear agrupamiento aleatorio (por ejem., inicio de k-medias)
2. Eliminar clusters de pocos elementos (redistribuir esos datos)
3. Calcular distancias medias intra-cluster
4. Calcular media ponderada de las distancias de 3
5. Si la distancia intra-cluster es suficientemente grande, o hay pocos clusters respecto a los deseados, se escinde el cluster
6. Calcular distancia inter-clusters
7. Si la distancia intercluster es suficientemente pequeña, fusionar clusters
8. Si no es la última iteración y no hay convergencia, volver a 3.

# Fuzzy clustering

## Idea básica

- ❑ Los métodos de agrupamiento clásicos suponen, al menos implícitamente, la hipótesis de que los objetos se partitionan en conjuntos disjuntos.
- ❑ Si los grupos son compactos y están bien separados esta es la mejor opción. Pero hay clusters cuyas fronteras no están muy claras, mal definidas o borrosas.



Los subconjuntos difusos (fuzzy sets) modelizan esta situación

# Fuzzy clustering

## Subconjunto difuso

Dado un conjunto  $X$ , un subconjunto difuso  $\mu$  es una aplicación

$$\mu : X \rightarrow [0, 1]$$

que asigna a cada elemento un valor de pertenencia

- Es una extensión de la teoría de conjuntos clásica, que serían las funciones que sólo toman el valor 0 (no está) o 1 (sí está)
- El valor de pertenencia no se interpreta como la probabilidad de que esté en el conjunto, si no como el grado de compatibilidad del elemento con el conjunto, entendido este como el resultado de una propiedad (o un conjunto de propiedades) expresadas de forma imprecisa

# Fuzzy clustering

## Agrupamiento difuso

Un agrupamiento difuso de un conjunto de objetos  $X$ , es una tupla de subconjuntos difusos de  $X$

$$(\mu_1, \mu_2, \dots, \mu_k)$$

donde cada conjunto difuso  $\mu_i : X \rightarrow [0, 1]$  mide la pertenencia de los objetos al cluster  $i$

Es habitual imponer la condición de **partición difusa o posibilística** (aunque no es estrictamente necesario)

$$\sum_{i=1}^k \mu_i(x) = 1 \text{ para todo } x \in X$$

# Fuzzy clustering

## Agrupamiento difuso

Este enfoque es muy útil cuando se intenta una interpretación de los grupos, ya que, en muchos casos, las descripciones de los grupos obtenidos en un problema concreto serán de tipo impreciso por serlo las etiquetas que los caracterizan

- Por ejemplo, coche de gama “alta”, “media”, “baja”

Normalmente los algoritmos difusos son adaptaciones de los algoritmos usuales al ambiente difuso

# Fuzzy c-means (Dunn, 1973)

---

## Algorithm 1 Algoritmo fuzzy c-means

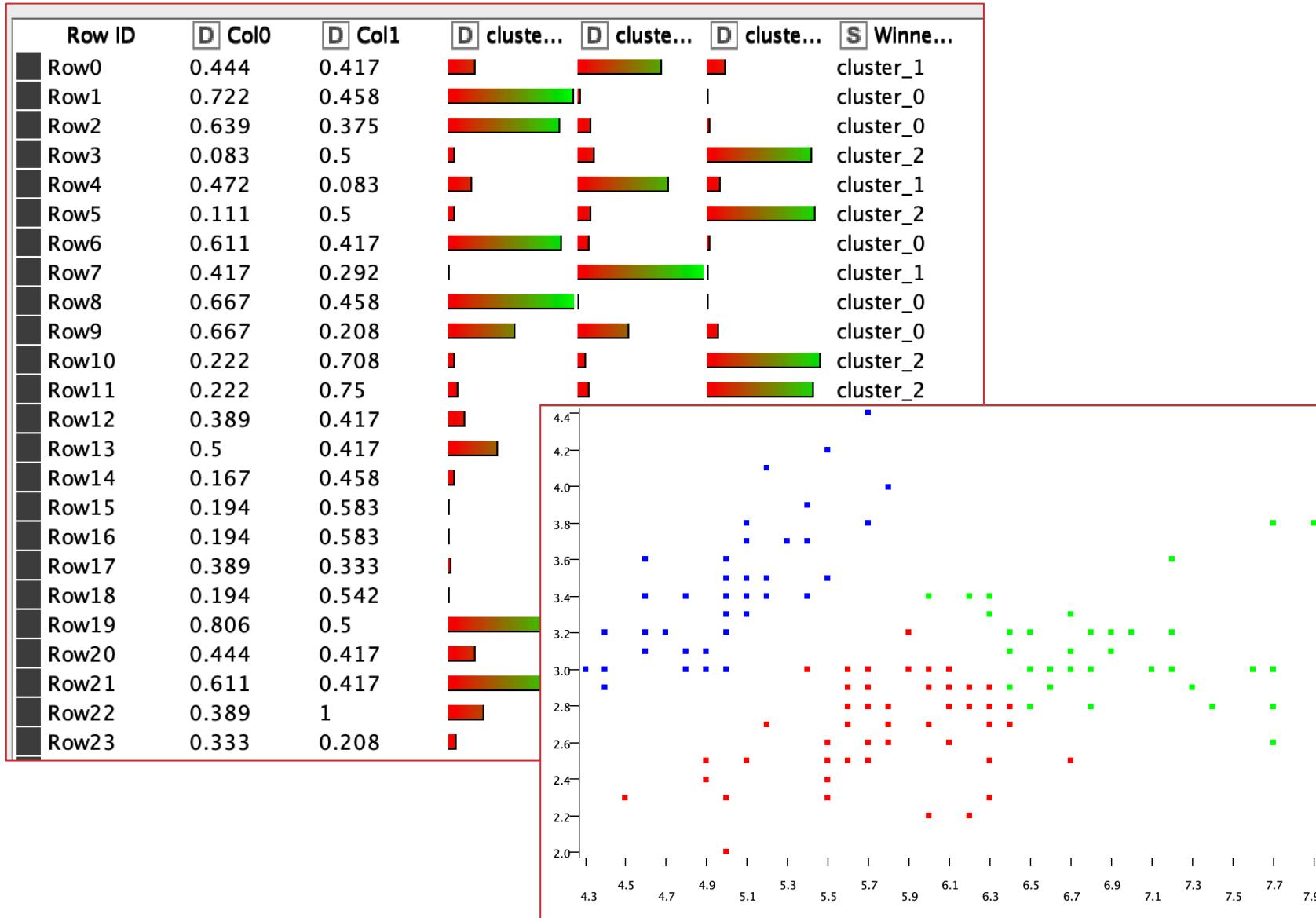
---

**Input:**  $X$  base de datos,  $k$  número de clusters,  $\epsilon$  umbral de convergencia

**Output:**  $\mu_1, \mu_2, \dots, \mu_k$  clusters difusos

- 1: Seleccionar de forma aleatoria  $k$  centroides
  - 2: Calcular clusters difusos con la fórmula  $\mu_i(x) = \frac{1}{\sum_{j=1}^k \left( \frac{d(x, c_i)}{d(x, c_j)} \right)^2}$
  - 3: Calcular suma del error cuadrático  $SSE = \sum_{x \in X} \sum_{i=1}^k \mu_i(x)^2 d(x, c_i)^2$
  - 4: **while**  $SSE$  cambie más de  $\epsilon$  en una iteración **do**
    - $\sum \mu_i(x)x$
    - 5: Recalcular centroides  $c_i = \frac{\sum_{x \in X} \mu_i(x)x}{\sum_{x \in X} \mu_i(x)}$
    - 6: Recalcular clusters difusos
    - 7: Recalcular  $SSE$
  - 8: **return**  $\mu_1, \mu_2, \dots, \mu_k$
-

# Fuzzy c-means (Dunn, 1973)



# Índice

- ❑ Concepto de clustering
- ❑ Distancias
- ❑ Clustering jerárquico
- ❑ Clustering basado en representantes
- ❑ **Clustering basado en densidad**
- ❑ Grid-based methods
- ❑ Clustering basado en modelos
- ❑ Evaluación/validación del clustering

# Density-based clustering

- ❑ En general, los métodos basados en representantes son buenos si la forma de los clusters es elipsoidal o, al menos, convexa
- ❑ Esto no siempre es así...



# Density-based clustering

- ❑ Los métodos basados en densidad consideran que los clusters son regiones densas del espacio de datos, separadas por regiones con poca densidad
- ❑ Por tanto, un cluster se define como el conjunto maximal de puntos conectados por densidad
- ❑ Estos métodos son capaces de determinar cluster con cualquier forma (no sólo convexos)

# DBSCAN

## □ Conceptos básicos

- Vecindario (bola) de una instancia/objeto

$$N_{\epsilon}(\mathbf{x}) = B_d(\mathbf{x}, \epsilon) = \{\mathbf{y} \mid \delta(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$$

distancia

radio de la bola

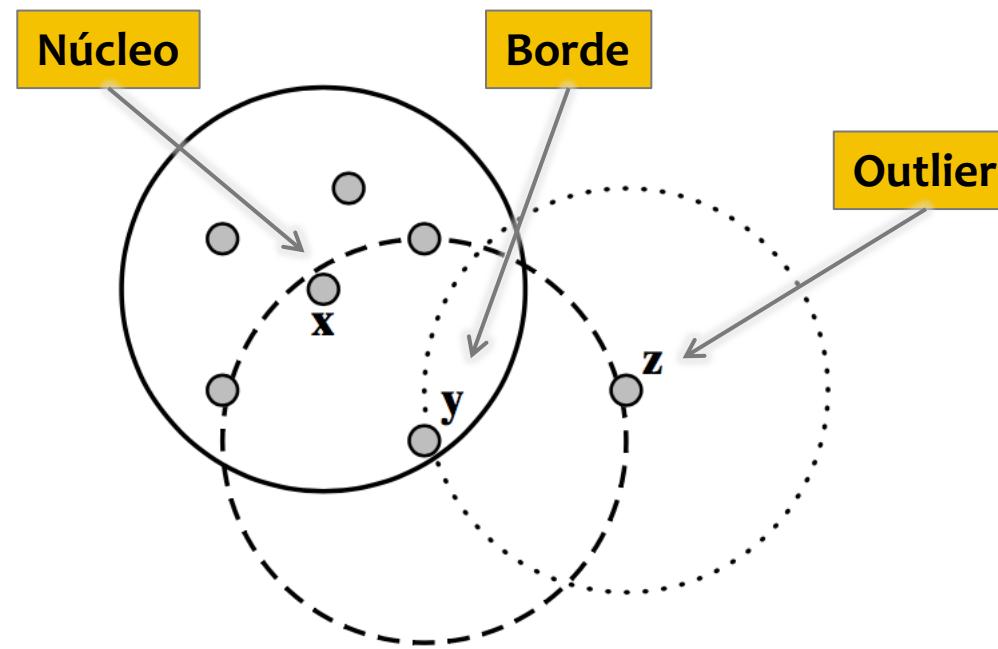
# DBSCAN

## □ Conceptos básicos

- Un **elemento núcleo** (core) es una instancia en cuyo vecindario hay al menos un mínimo preestablecido de elementos
- Un **elemento borde** es (border point) una instancia cuyo vecindario tiene menos elementos que ese mínimo pero está en el vecindario de un elemento núcleo
- Si no es un elemento núcleo o border, se dice que dicho elemento es un **elemento ruido** o **outlier**

# DBSCAN

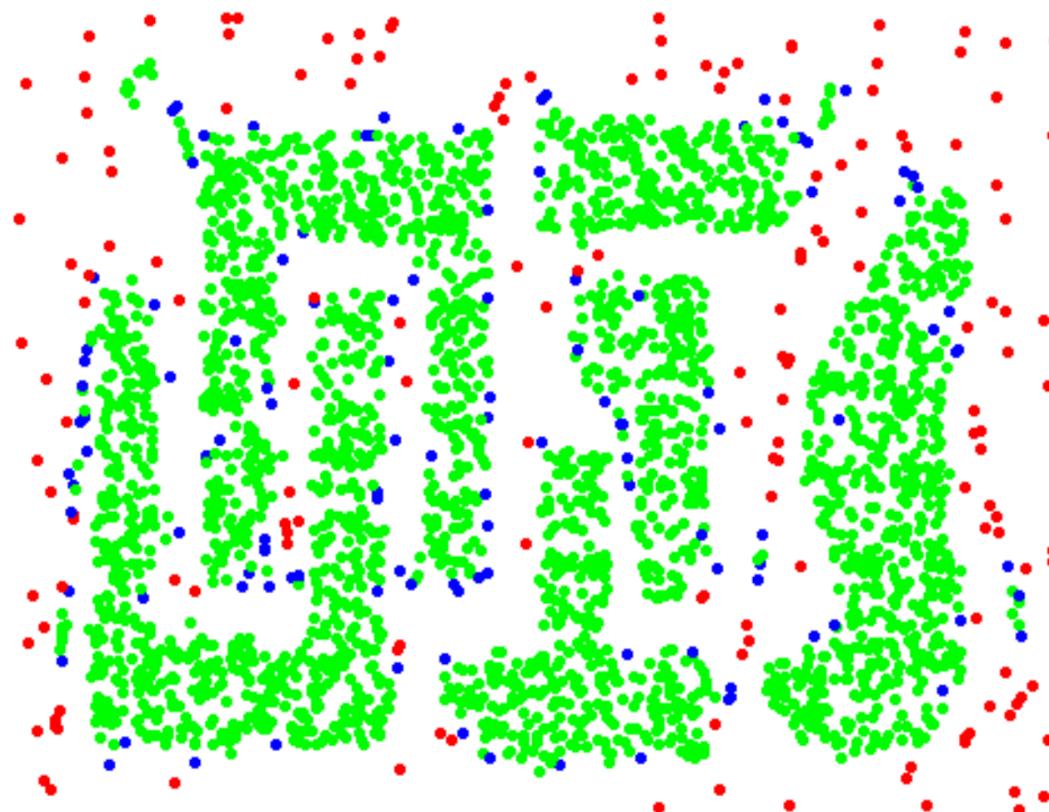
## □ Conceptos básicos



**cinco elementos para ser un núcleo!**

# DBSCAN

## □ Conceptos básicos

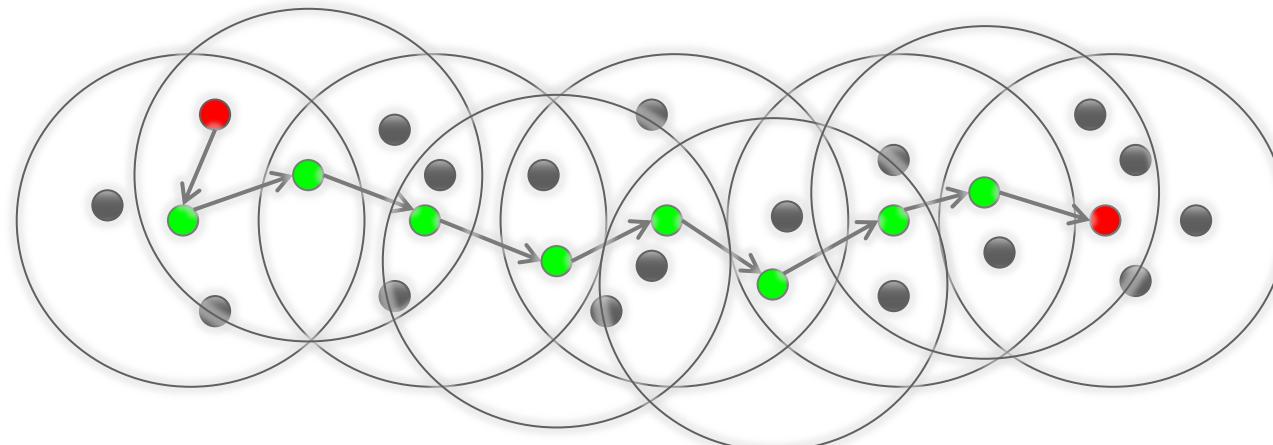


Minpoints=4  
Epsilon=10  
**Núcleo**  
**Borde**  
**Outlier**

# DBSCAN

## □ Conceptos básicos

- Un objeto A es **directamente densamente alcanzable** de un objeto B si B es un núcleo y A está en su vecindario
- Un objeto A es **densamente alcanzable** de un objeto B si existe una secuencia  $x_1, x_2, \dots, x_n$  con  $x_1=A$ ,  $x_n=B$  y  $x_i$  es directamente densamente alcanzable de  $x_{i-1}$



# DBSCAN

---

## ALGORITHM

## Density-based Clustering Algorithm

---

**DBSCAN ( $\mathbf{D}, \epsilon, minpts$ ):**

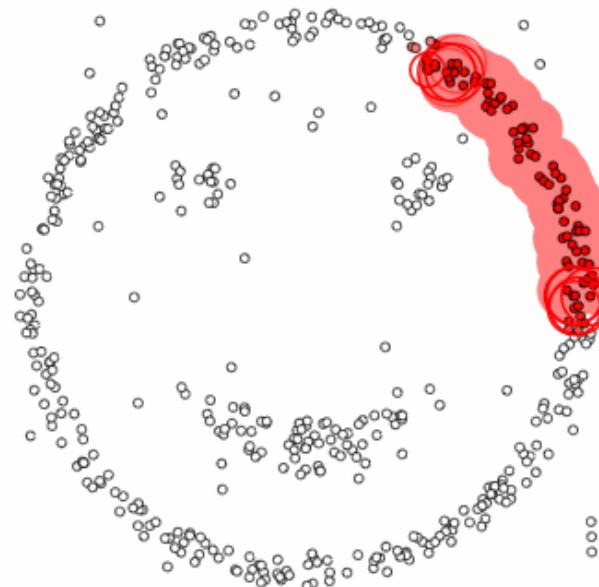
- 1  $Core \leftarrow \emptyset$
- 2 **foreach**  $\mathbf{x}_i \in \mathbf{D}$  **do** // Find the core points
  - 3     Compute  $N_\epsilon(\mathbf{x}_i)$
  - 4      $id(\mathbf{x}_i) \leftarrow \emptyset$  // cluster id for  $\mathbf{x}_i$
  - 5     **if**  $N_\epsilon(\mathbf{x}_i) \geq minpts$  **then**  $Core \leftarrow Core \cup \{\mathbf{x}_i\}$
- 6      $k \leftarrow 0$  // cluster id
- 7     **foreach**  $\mathbf{x}_i \in Core$ , such that  $id(\mathbf{x}_i) = \emptyset$  **do**
  - 8          $k \leftarrow k + 1$
  - 9          $id(\mathbf{x}_i) \leftarrow k$  // assign  $\mathbf{x}_i$  to cluster id  $k$
  - 10         DENSITYCONNECTED ( $\mathbf{x}_i, k$ )
- 11      $\mathcal{C} \leftarrow \{C_i\}_{i=1}^k$ , where  $C_i \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = i\}$
- 12      $Noise \leftarrow \{\mathbf{x} \in \mathbf{D} \mid id(\mathbf{x}) = \emptyset\}$
- 13      $Border \leftarrow \mathbf{D} \setminus (Core \cup Noise)$
- 14     **return**  $\mathcal{C}, Core, Border, Noise$

**DENSITYCONNECTED ( $\mathbf{x}, k$ ):**

- 15     **foreach**  $\mathbf{y} \in N_\epsilon(\mathbf{x})$  **do**
  - 16          $id(\mathbf{y}) \leftarrow k$  // assign  $\mathbf{y}$  to cluster id  $k$
  - 17         **if**  $\mathbf{y} \in Core$  **then** DENSITYCONNECTED ( $\mathbf{y}, k$ )

Complejidad cuadrática en el peor caso

# DBSCAN



epsilon = 1.00  
minPoints = 4

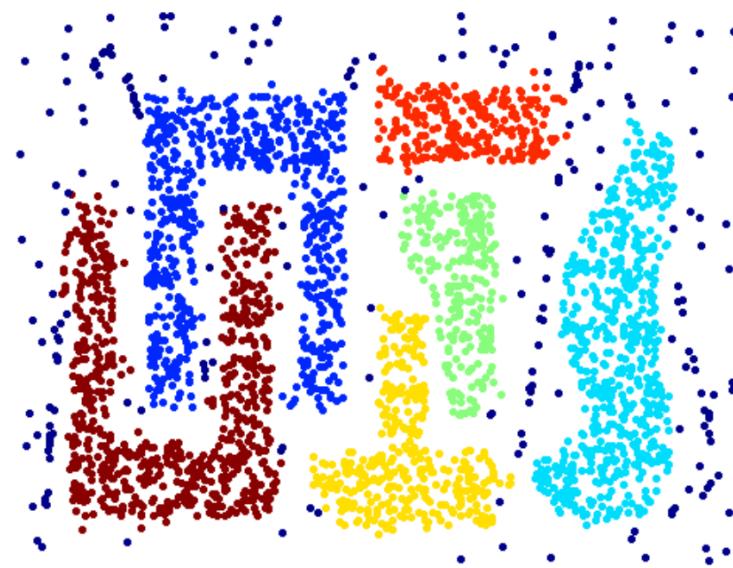
Restart



Pause

# DBSCAN

Cuando el algoritmo funciona bien...



- Resistente al ruido
- Clusters de cualquier forma y tamaño

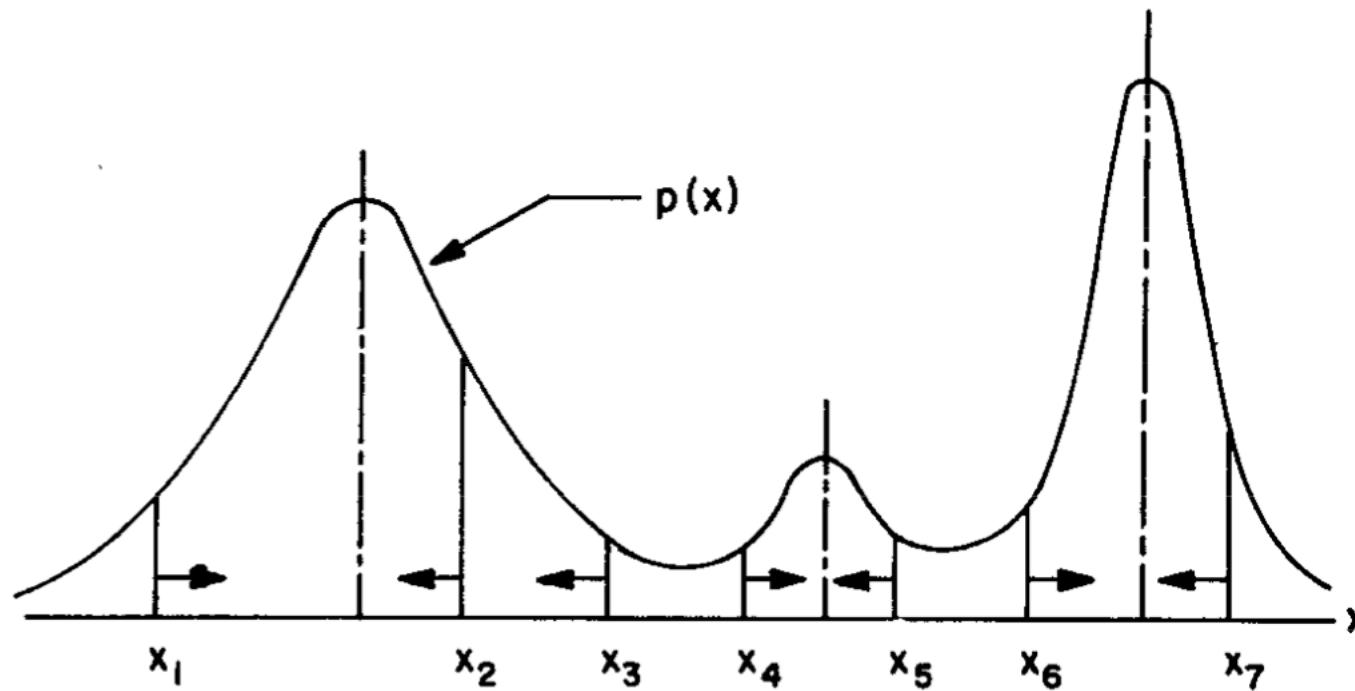
# DBSCAN

Problemas que presenta:

- ❑ Muy dependiente de los parámetros
  - diámetro de los entornos
  - mínimo de puntos para ser núcleo
- ❑ Difícil de establecer los parámetros correctos
- ❑ No lleva bien que un mismo cluster presente zonas con diferentes densidades

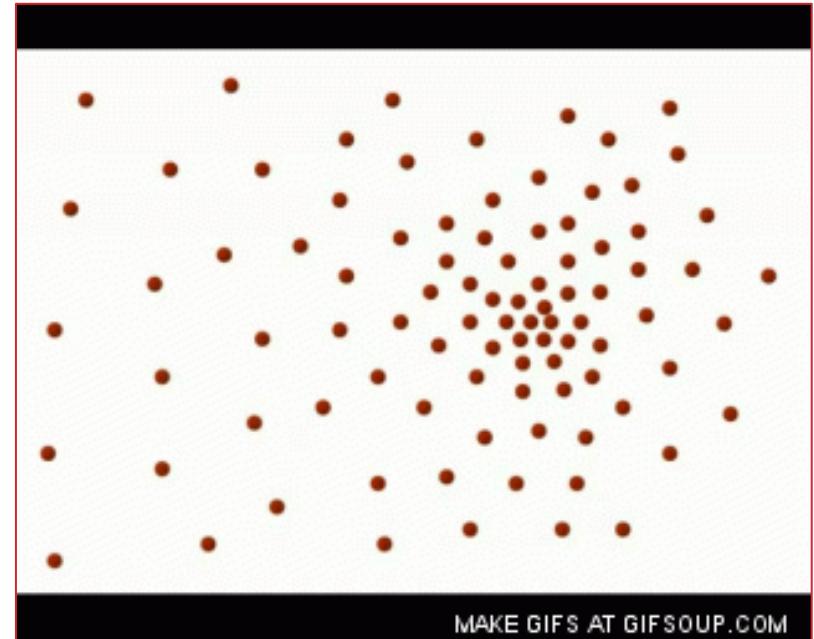
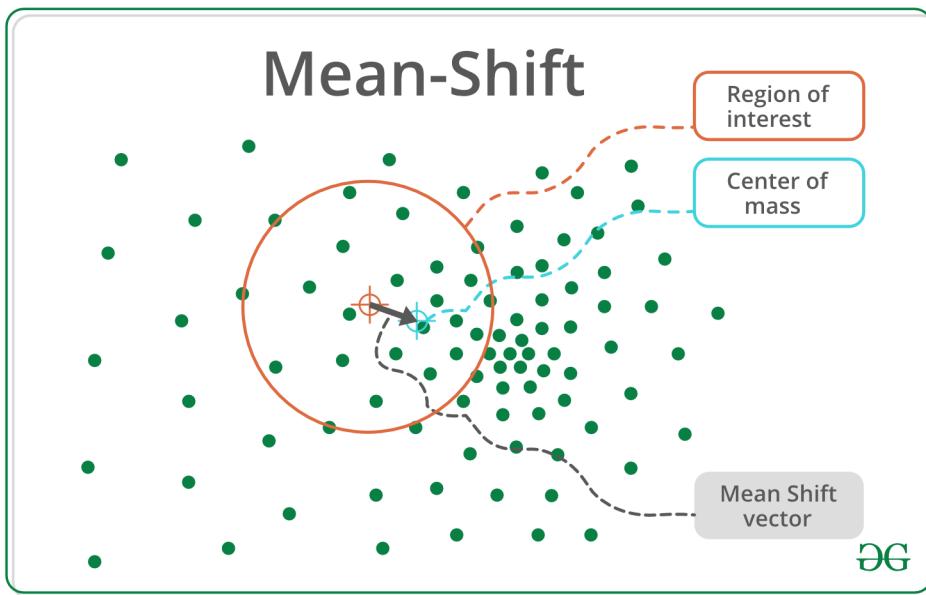
# Mean Shift (Fukunaga, Hostetler, 1975)

Se basa en un proceso iterativo en que cada punto debe dirigirse a su óptimo local más próximo. Los puntos con óptimos locales suficientemente cerca, pertenecen al mismo cluster

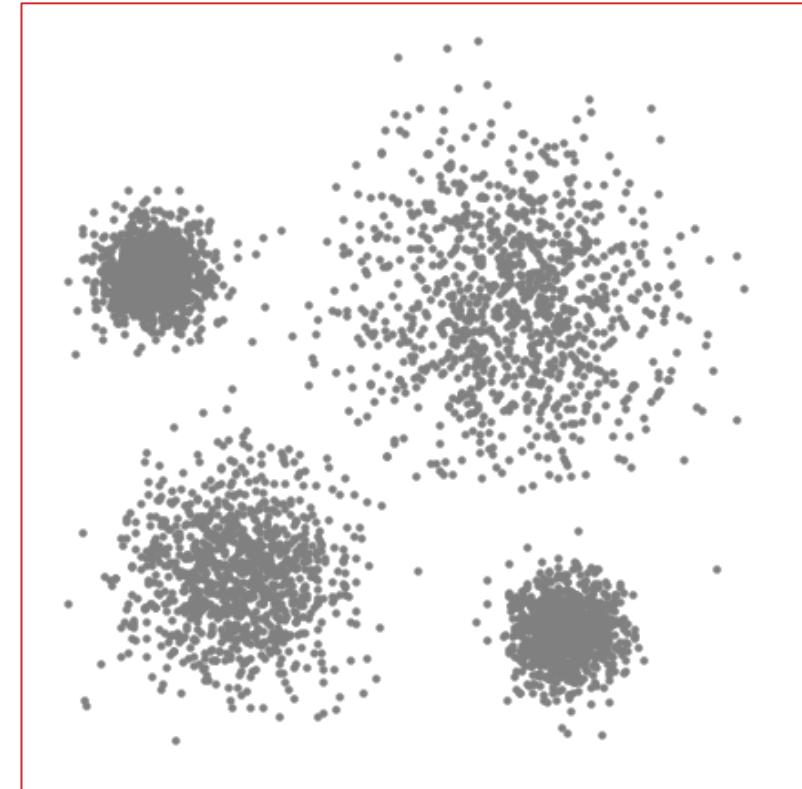
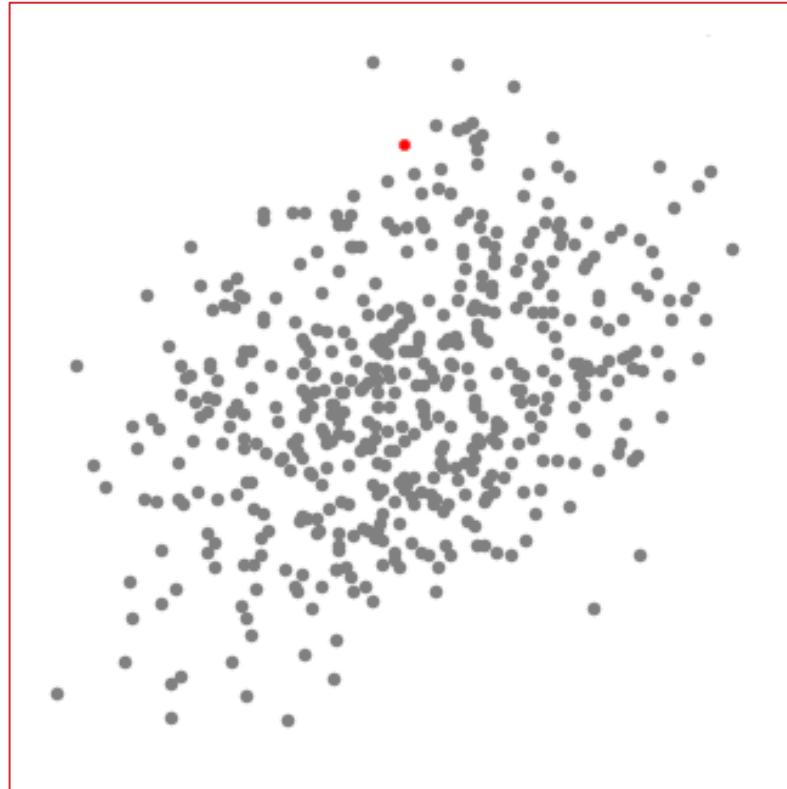


# Mean Shift (Fukunaga, Hostetler, 1975)

La forma de dirigirse al óptimo es buscando las zonas de mayor densidad del entorno de cada punto



# Mean Shift (Fukunaga, Hostetler, 1975)



# Mean Shift (Fukunaga, Hostetler, 1975)

Primero se busca una función de densidad adecuada (kernel density estimation, KDE). Normalmente, una gaussiana

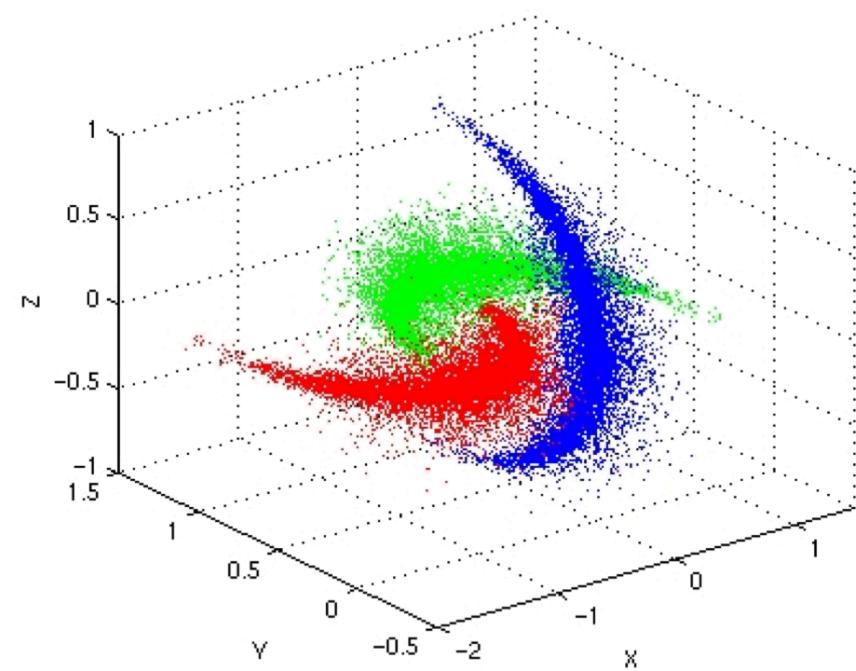
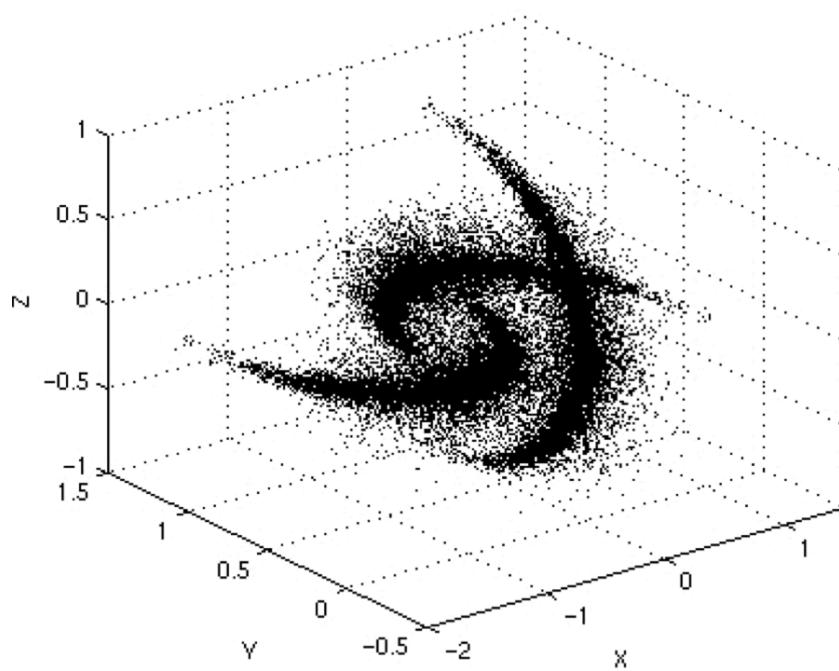
$$k(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^t \Sigma^{-1} (x-\mu)}$$

El movimiento de los puntos se realiza en la dirección del gradiente

$$x_{i+1} = x_i - \alpha \nabla \log k(x_i)$$

# Mean Shift (Fukunaga,Hostetler,1975)

**Ventajas:** No se necesita establecer el número de clusters y reconoce clusters de cualquier forma



# Otros algoritmos

**OPTICS** (Ordering Points to Identify the Clustering Structure)

M. Ankerst, M. M. Breunig, H-P. Kriegel, J. Sander, OPTICS: Ordering Points To Identify the Clustering Structure. ACM SIGMOD international conference on Management of data. ACM Press. 49–60

**DENCLUE**: DENsity-based CLUstEring (Hinneburg & Keim, KDD'1998)

**CLIQUE**: Clustering in QUEst (Agrawal et al., SIGMOD'1998)

**SNN** (Shared Nearest Neighbor) density-based clustering (Ertöz, Steinbach & Kumar, SDM'2003)

Trabajo evaluable (3 personas, 30 minutos). Clustering basado en densidad. Considera la referencia anterior y realiza un trabajo explicando la motivación del algoritmo OPTICS, su funcionamiento y presentando algunos ejemplos

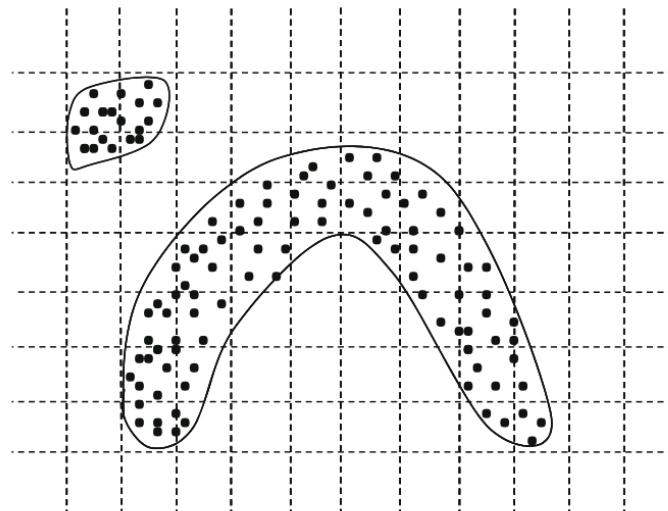
# Índice

- ❑ Concepto de clustering
- ❑ Distancias
- ❑ Clustering jerárquico
- ❑ Clustering basado en representantes
- ❑ Clustering basado en densidad
- ❑ **Grid-based methods**
- ❑ Clustering basado en modelos
- ❑ Evaluación/validación del clustering

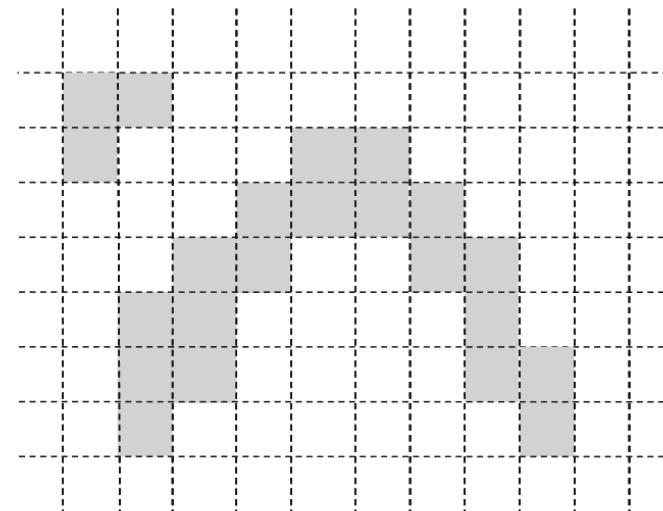
# Grid-based methods

Los datos se discretizan en intervalos

Los datos pertenecientes a hipercubos (suficientemente densos) adyacentes pertenecen al mismo cluster



(a) Data points and grid



(b) Agglomerating adjacent grids

# Grid-based methods

**Algorithm** *GenericGrid*(Data:  $\mathcal{D}$ , Ranges:  $p$ , Density:  $\tau$  )  
**begin**

    Discretize each dimension of data  $\mathcal{D}$  into  $p$  ranges;  
    Determine dense grid cells at density level  $\tau$ ;  
    Create graph in which dense grids are connected if they are adjacent;  
    Determine connected components of graph;  
    **return** points in each connected component as a cluster;

**end**

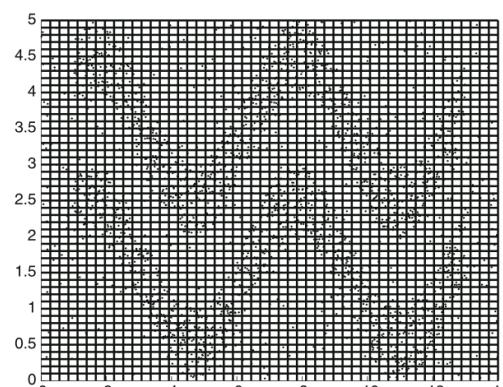
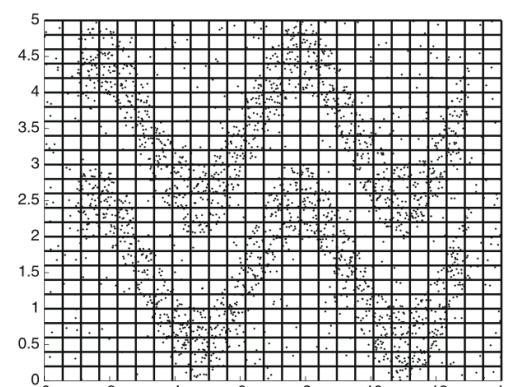
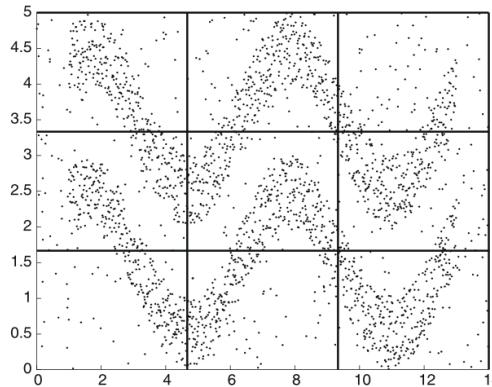
# Grid-based methods

## Ventajas

- ❑ No hay que establecer el número de clusters
- ❑ Construye clusters con cualquier forma

## Inconvenientes:

- ❑ Depende en exceso de la longitud de los intervalos y del umbral de densidad
- ❑ No se lleva bien con dimensiones altas
  - 100 dimensiones + 2 intervalos por dimensión ->  $2^{100}$  hipercubos



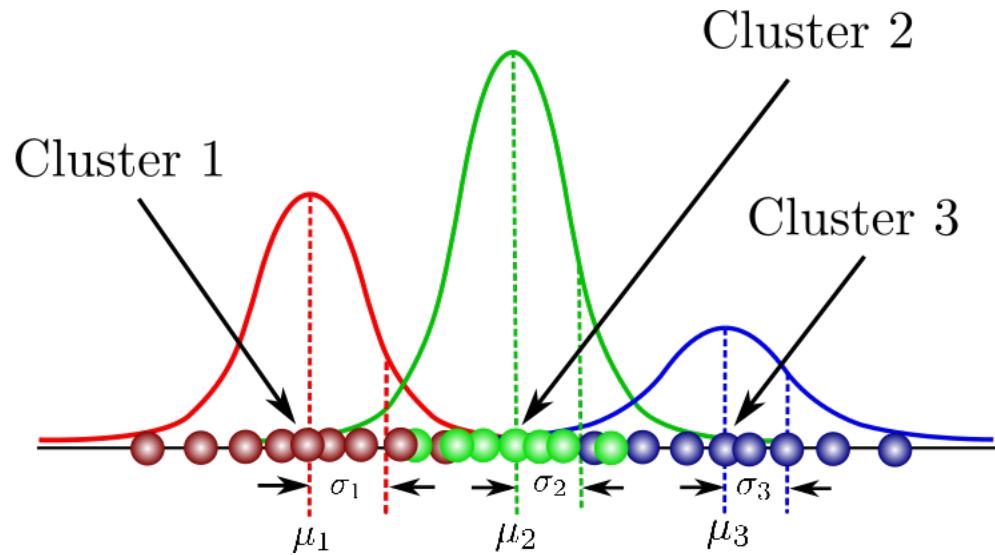
# Índice

- ❑ Concepto de clustering
- ❑ Distancias
- ❑ Clustering jerárquico
- ❑ Clustering basado en representantes
- ❑ Clustering basado en densidad
- ❑ Grid-based methods
- ❑ **Clustering basado en modelos**
- ❑ Evaluación/validación del clustering

# Model-based clustering

Se asume que los datos siguen un modelo preestablecido, normalmente estadístico. El más conocido, **GMM** (Gaussian Mixture Model)

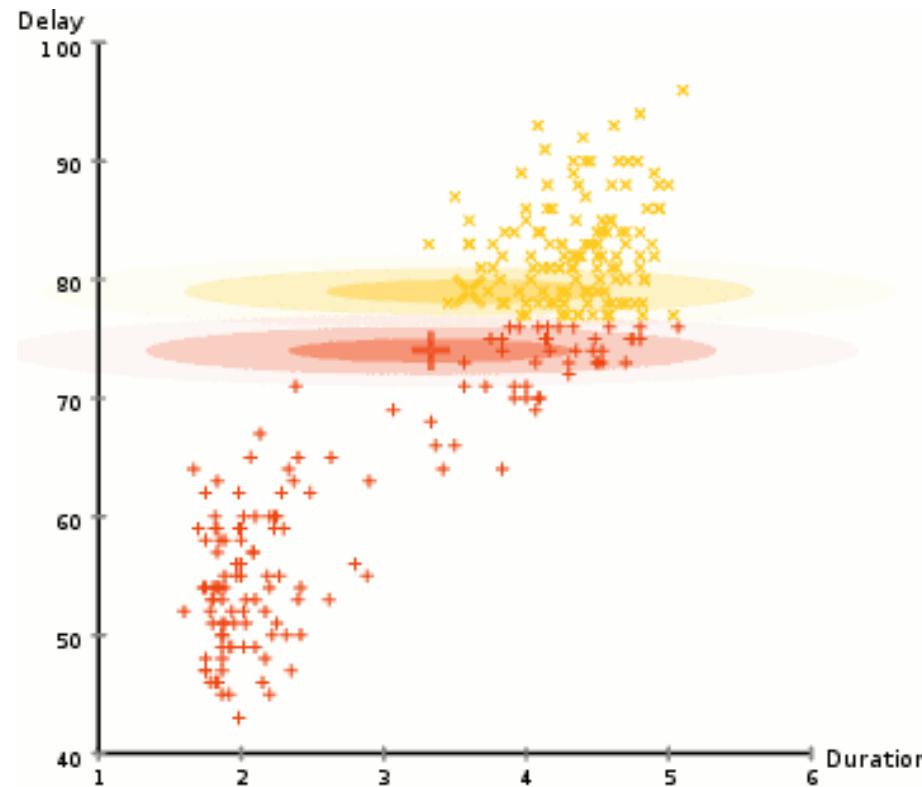
Supone que los datos provienen de distribuciones gaussianas que se superponen



Se trata de estimar los parámetros (media y varianza)

# Model-based clustering

Para el cálculo de los parámetros se puede utilizar el algoritmo **EM** (Expectation-Maximization)



# Índice

- ❑ Concepto de clustering
- ❑ Distancias
- ❑ Clustering jerárquico
- ❑ Clustering basado en representantes
- ❑ Clustering basado en densidad
- ❑ Grid-based methods
- ❑ Clustering basado en modelos
- ❑ **Evaluación/validación del clustering**

# Validación

¿Cómo se puede evaluar la calidad del clustering?

Es difícil de evaluar, no hay un valor numérico que diga si un modelo de clustering se acerca a la realidad. Ya que, principio, no sabemos nada de los datos. Es **aprendizaje no supervisado**.

# Validación

Pero necesitamos evaluar para, por ejemplo,

- Evitar descubrir clusters donde sólo hay ruido
- Comparar dos agrupamientos diferentes
- Comparar dos algoritmos de agrupamiento
- Determinación de la tendencia de agrupamiento de un conjunto de datos, es decir , distinguir si la estructura no aleatoria realmente existe en los datos.
- Determinación del número "correcto" de clusters
- ...

# Medidas de evaluación

1. **No supervisadas.** Sin utilizar ninguna información externa. También se llaman internas

Por ejemplo, el error cuadrático medio

$$\frac{1}{n} \sum_{i=1}^k \sum_{c \in C_k} d(c, d_k)^2$$

como en k-medias

# Medidas de evaluación

1. **No supervisadas.** Sin utilizar ninguna información externa. También se llaman internas

**Medidas de cohesión.** Cómo de relacionados están los objetos de un cluster?

$$\text{SSE} = \sum_{i=1}^k \sum_{a \in C_i} d(a, \text{centro}[i])^2$$

**Medidas de cohesión.** Cómo de bien separados están los objetos de distintos clusters?

$$\sum_{i=1}^k |C_i| d(\text{centro}, \text{centro}[i])^2$$

# Medidas de evaluación

**Coeficiente de silueta (Silhouette Coefficient).** Combina las ideas de separación y cohesión, para cada objeto  $a \in C_i$

1. Se calcula su distancia media a los elementos de su cluster

$$a_c = \frac{\sum_{b \in C_i, a \neq b} d(a, b)}{|C_i| - 1}$$

2. Se calcula la media de su distancia a los elementos de cada uno de los otros clusters, y se selecciona el mínimo

$$b_j = \frac{\sum_{b \in C_j} d(a, b)}{|C_j|} \quad a_d = \min\{b_j \mid i \neq j\}$$

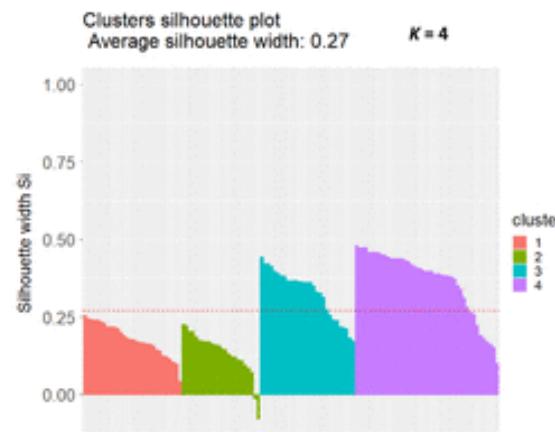
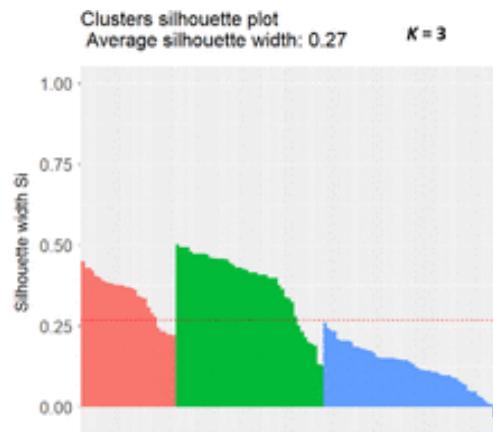
3. El coeficiente de silueta viene dado por

$$s_a = \frac{a_d - a_c}{\max(a_c, a_d)}$$

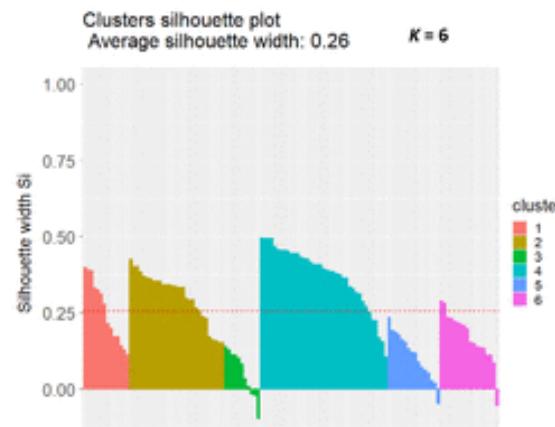
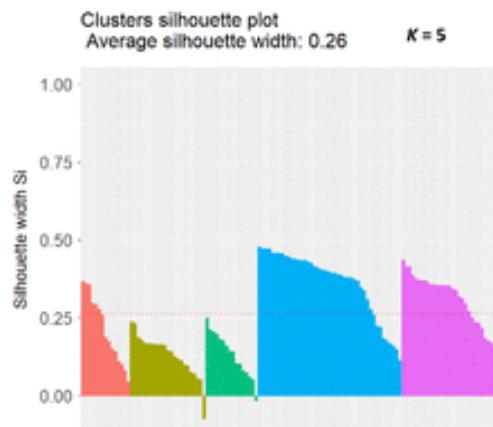
# Medidas de evaluación

## Coeficiente de silueta (Silhouette Coefficient)

El coeficiente varía entre -1 y 1. Es deseable un coeficiente positivo y cerca del 1



El coeficiente de silueta de un agrupamiento es la media de los correspondientes coeficientes

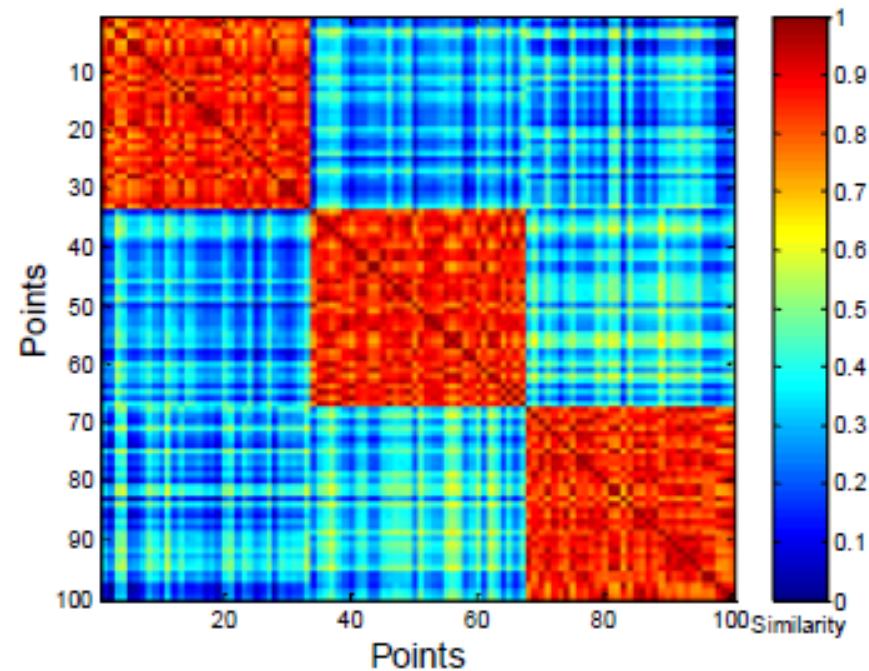
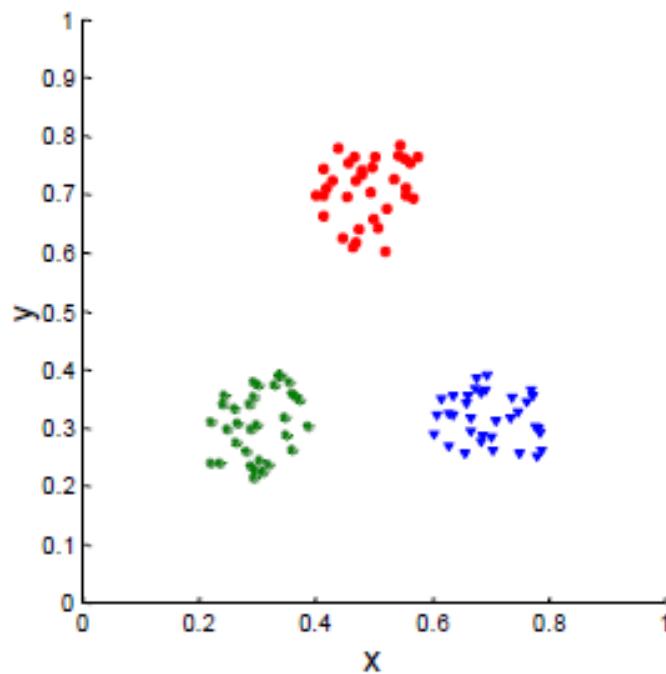


[link](#)

# Medidas de evaluación

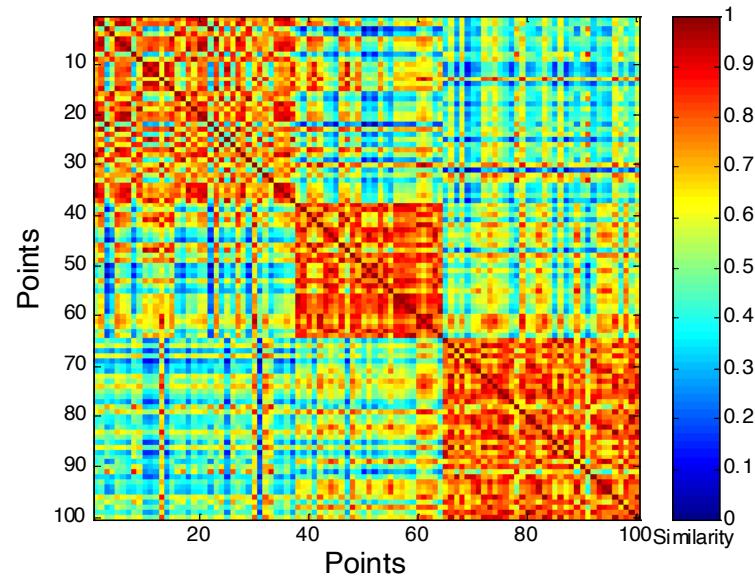
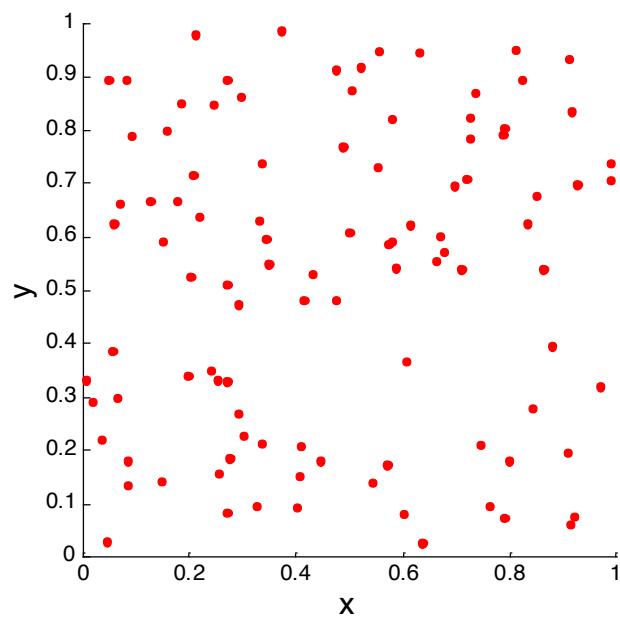
## Matriz de similitud

Ordenamos los datos en la matriz de similitud con respecto a los clusters en los que quedan los datos e inspeccionamos visualmente...



# Medidas de evaluación

Cluster en datos aleatorios (K-medias)



# Medidas de evaluación

2. **Supervisadas.** Con algún conocimiento externo. También se llaman externas

**Orientadas a clasificación.** Miden cómo se ajusta la partición obtenida en un agrupamiento a una clasificación previamente dada.

Sean  $\{G_1, \dots, G_k\}$  los grupos obtenidos y  $\{C_1, \dots, C_t\}$  las clases a comparar, sea  $m$  el número de puntos. Definimos:

$$p_{ij} = \frac{m_{ij}}{m_i} \text{ para todo } i \in \{1, \dots, k\}, j \in \{1, \dots, t\}$$

donde  $m_{ij}$  es el número de objetos que hay de la clase  $C_j$  en el grupo  $G_i$  y  $m_i$  es el número de ítems que hay en el grupo  $G_i$ .  $p_{ij}$  es la probabilidad de que un objeto de  $G_i$  pertenezca a  $C_j$ .

# Medidas de evaluación

- Entropía de un cluster       $e_i = - \sum_{j=1}^t p_{ij} \log_2 p_{ij} \quad \forall i \in \{1, \dots, k\}$
- Entropía total       $e = \frac{\sum_{i=1}^k m_i e_i}{m}$
- Puridad de un cluster       $p_i = \max\{p_{ij} \mid j \in \{1, \dots, t\}\}$
- Puridad total       $p = \frac{\sum_{i=1}^k m_i p_i}{m}$
- Precisión       $\text{precision}(i, j) = p_{ij}$
- Recall       $\text{recall}(i, j) = m_{ij}/n_j$  donde  $n_j = \#C_j$
- F-medida       $F(i, j) = \frac{2 \text{precision}(i, j) \text{recall}(i, j)}{\text{precision}(i, j) + \text{recall}(i, j)}$

# Medidas de evaluación

2. **Supervisadas.** Con algún conocimiento externo. También se llaman externas

**Orientadas a similaridad.** La idea básica se construir las matrices de incidencia de los agrupamientos

$$IG_{ab} = \begin{cases} 1 & \text{si } a, b \text{ en el mismo cluster} \\ 0 & \text{en otro caso} \end{cases}$$

$$IC_{ab} = \begin{cases} 1 & \text{si } a, b \text{ en el mismo cluster} \\ 0 & \text{en otro caso} \end{cases}$$

y establecer medidas de coincidencia entre ambas

# Evaluación

Otra alternativa, calcular una matriz de confusión a partir de las matrices

	En cluster “verdadero”	No en cluster “verdadero”
En cluster	True Positive (TP)	False Negative (FN)
No en cluster	False Positive (FP)	True Negative (TN)

Y calcular alguna medida sobre ella

$$\frac{TP + TN}{TP + TN + FN + FP}$$

Estadístico de Rand/  
Exactitud

$$\frac{TP}{TN + FN + FP}$$

Coeficiente de Jaccard

# Medidas de evaluación

Clustering

	g 1	g 2	g 3	g 4	g 5
g 1	1	1	1	0	0
g 2	1	1	1	0	0
g 3	1	1	1	0	0
g 4	0	0	0	1	1
g 5	0	0	0	1	1

Groundtruth

	g 1	g 2	g 3	g 4	g 5
g 1	1	1	0	0	0
g 2	1	1	0	0	0
g 3	0	0	1	1	1
g 4	0	0	1	1	1
g 5	0	0	1	1	1

	Same Cluster	Different Cluster
Same Cluster	9	4
Different Cluster	4	8

$$\text{Exactitud} = \frac{17}{25} = 0.68$$

# Más aspectos sobre el clustering

## Sobre tipos de datos

- Algoritmos específicos para clustering de documentos, data stream, imágenes, datos categóricos, sonidos, grafos,...

## Otras modificaciones realizar agrupamientos

- Clustering con restricciones
- Clustering con dimensiones altas
- Subspace clustering
- Clustering semisupervisado
- Cluster ensembles
- ...

# Trabajos evaluables

3 personas, 30 minutos. **Network clustering.** Bibliografía:

- J. Han, M. Kamber and J. Pei. Data Mining, Second Edition: Concepts and Techniques. Morgan Kaufmann, 2006. **Capítulo 11**
- C.C. Agrawal and C. Reddy, Data Clustering. Algorithms and Applications, Chapman & Hall, 2014. **Capítulo 17**

2 personas, 20 minutos. **Grid-based clustering.** Bibliografía:

- C.C. Agrawal and C. Reddy, Data Clustering. Algorithms and Applications, Chapman & Hall, 2014. **Capítulo 6**

2 personas, 20 minutos. **Spectral clustering.** Bibliografía:

- C.C. Agrawal and C. Reddy, Data Clustering. Algorithms and Applications, Chapman & Hall, 2014. **Capítulo 8**

2 personas, 20 minutos. **High-dimensional clustering.** Bibliografía:

- C.C. Agrawal and C. Reddy, Data Clustering. Algorithms and Applications, Chapman & Hall, 2014. **Capítulo 9**

# Bibliografía

- J. H. Orallo, M. J. Ramírez Quintana, C. F. Ramírez. Introducción a la Minería de Datos. Pearson, 2004. **Capítulo 16.**
- J. Han, M. Kamber and J. Pei. Data Mining, Second Edition: Concepts and Techniques. Morgan Kaufmann, 2006. **Capítulos 10,11.**
- Mohammed J. Zaki, Wagner Meira, Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, May 2014. **Capítulos 13,14,17.**
- C.C. Agrawal and C. Reddy, Data Clustering. Algorithms and Applications, Chapman & Hall, 2014
- S. T. Wierzchoń and M. A. Kłopotek , Modern Algorithms of Cluster Analysis, Studies on Big Data 34, Springer, 2018.
- P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining Addison-Wesley, 2006. Capítulos 8,9.

# Bibliografía

- ❑ C. Aggarwal, Data Mining: The textbook, Springer, 2015.
- ❑ Anil K. Jain, M. Narasimha Murty & Patrick J. Flynn: Data Clustering: A Review. ACM Computing Surveys, Vol. 31, No. 3, pages 264-323, September 1999.
- ❑ L. Kaufman and P. J. Rousseeuw, Clustering Large Data Sets, in: Pattern Recognition in Practice II (1986), 425-435.
- ❑ <https://educlust.dbvis.de/#> (plataforma para docencia del clustering)

Algunas transparencias y gráficos tomados de:

- <http://elvex.ugr.es/idbis/dm/>