Aprendizaje Automático

Segundo Cuatrimestre de 2016

Aprendizaje No Supervisado

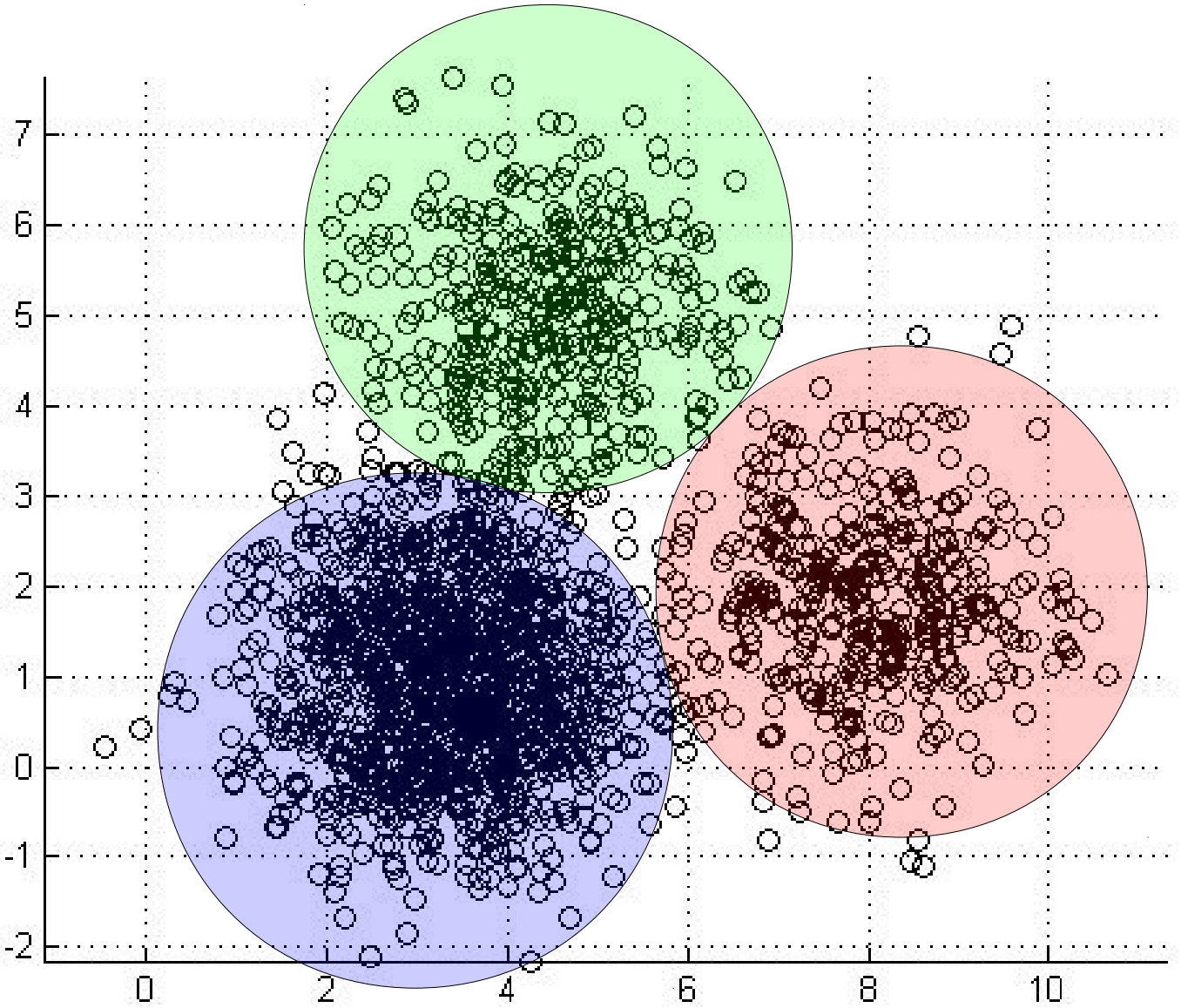
DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Supervisado vs. No Supervisado

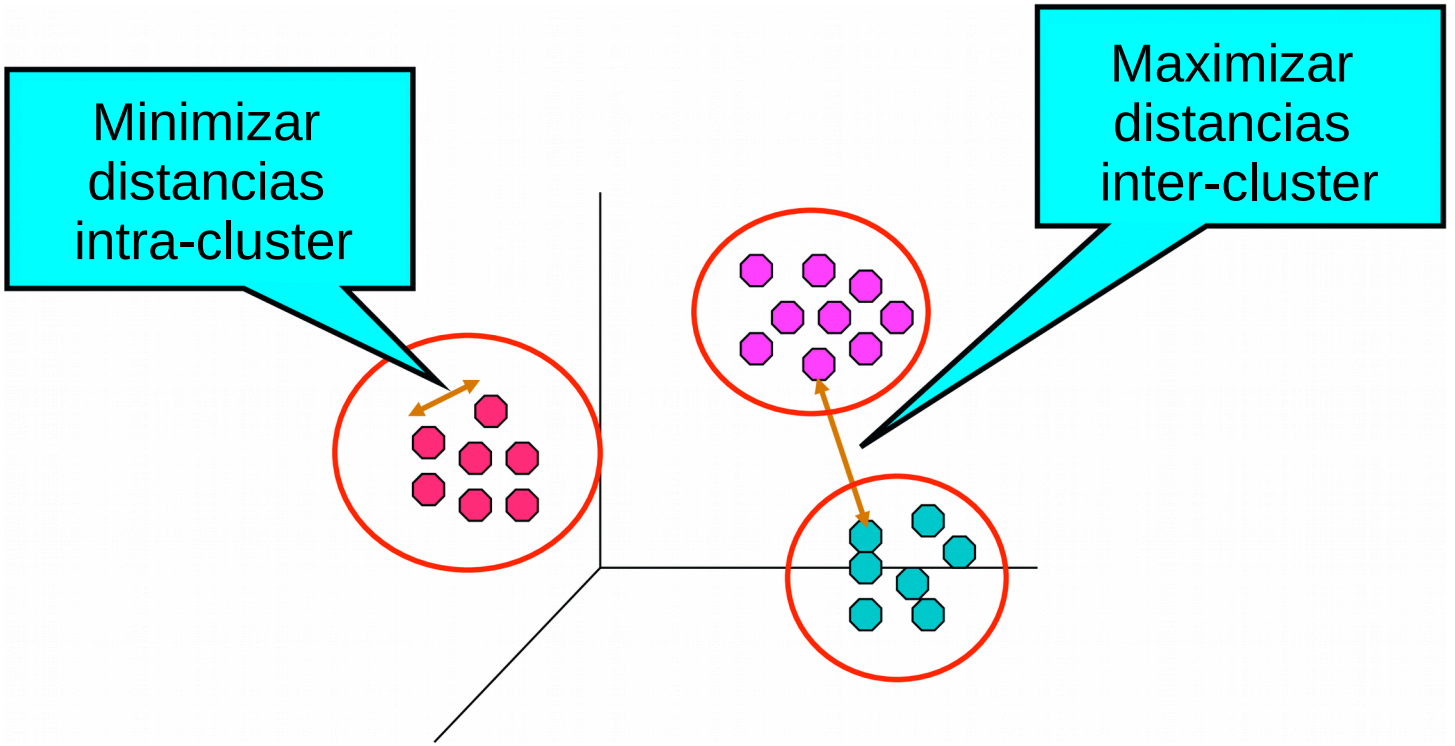
- Aprendizaje Supervisado
  - Clasificación y regresión.
  - Requiere instancias etiquetadas para entrenamiento.
- Aprendizaje No Supervisado
  - Clustering: particionar los datos en grupos cuando no hay categorías/clases disponibles.
  - Sólo requiere instancias, pero no etiquetas.
  - Sirve para entender y resumir los datos.

Clustering



Clustering

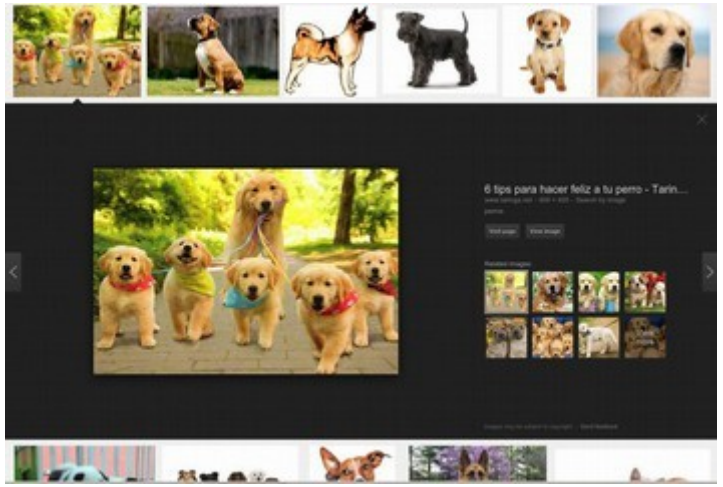
- **Objetivo:** Encontrar grupos de instancias tales que las instancias en un cluster sean similares entre sí, y diferentes de las instancias en otros clusters.



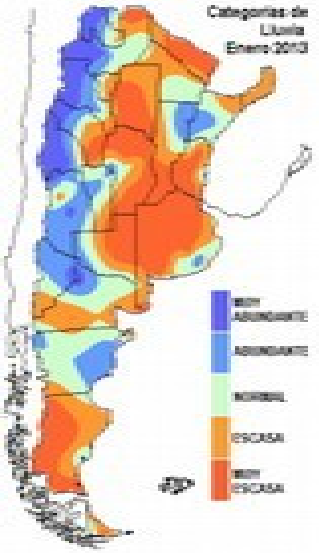
http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

Clustering: Aplicaciones

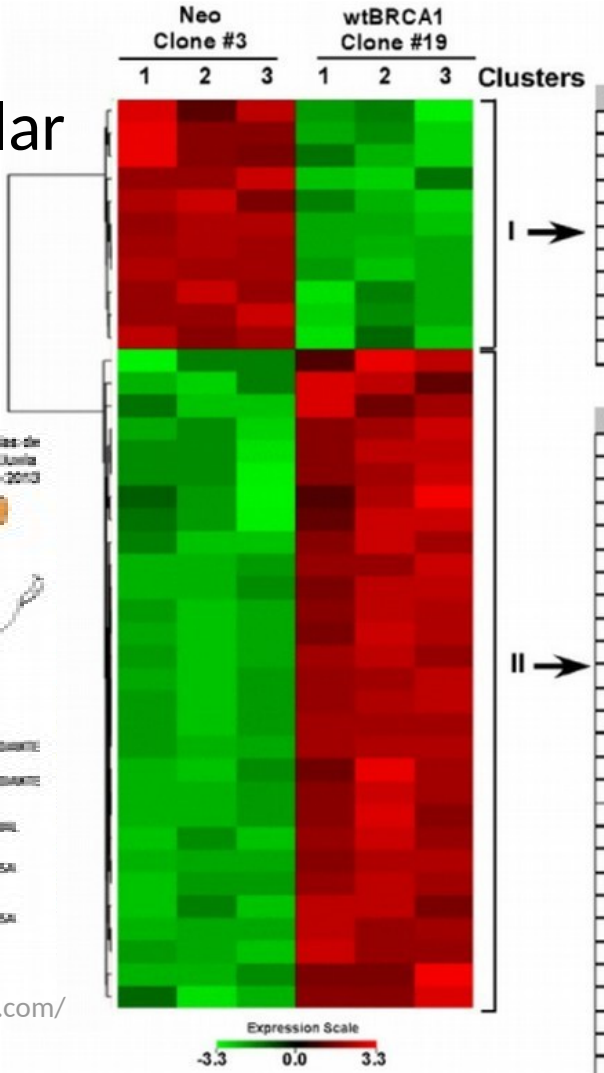
- Agrupar genes y proteínas con similar funcionalidad.
- Reducir el tamaño de conjuntos de datos grandes (ej: lluvias).



https://images.google.com/



http://www.agrositio.com/



Cluster I

Symbol	Fold	p-value
CCND2	-122.5	0.00000
CDKN2A	-21.4	0.00006
TNF	-8.8	0.00018
FZD8	-8.0	0.00029
FGF2	-6.8	0.00014
IL13RA2	-6.1	0.00025
BMP7	-6.1	0.00004
CDKN1A	-5.3	0.00001
FOXE1	-5.0	0.00001
MMP1	-4.9	0.00064
JAK1	-3.1	0.00036

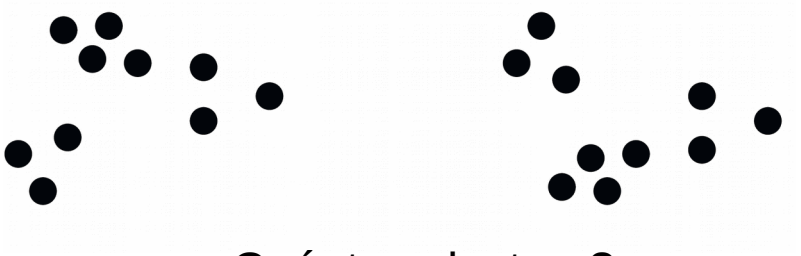
Cluster II

Symbol	Fold	p-value
EPSTI1	3.4	0.00038
TFF1	3.5	0.00008
FGF5	3.8	0.00027
TGFBR1	3.9	0.00020
NQO1	3.9	0.00001
EGFR	4.1	0.00004
CDKN2C	4.3	0.00027
CD24	4.6	0.00027
MGST2	4.8	0.00006
FZD7	5.2	0.00012
FZD1	5.6	0.00006
TGFBR3	5.7	0.00045
FOXG1B	6.4	0.00016
FZD4	6.8	0.00002
BCAR3	7.0	0.00001
FOXF2	9.2	0.00002
SMAD6	10.7	0.00000
INHBB	11.6	0.00001
PRKCA	16.0	0.00000
CCNA1	16.3	0.00023
GLRX	17.0	0.00021
TGFA	17.5	0.00001
GPX3	18.7	0.00006
MGST1	23.1	0.00000
FOXA2	36.5	0.00022
FOXP1	45.6	0.00000
FST	51.6	0.00002
TGFB2	70.8	0.00000
FGF13	71.8	0.00000

http://openi.nlm.nih.gov/detailedresult.php?img=3365892_pone.0037697.g002&req=4

- Agrupar documentos para explorarlos más rápido (ej: Google Images).

“Cluster” es un concepto ~~claro~~ ambiguo!



¿Cuántos clusters?

Dos clusters.

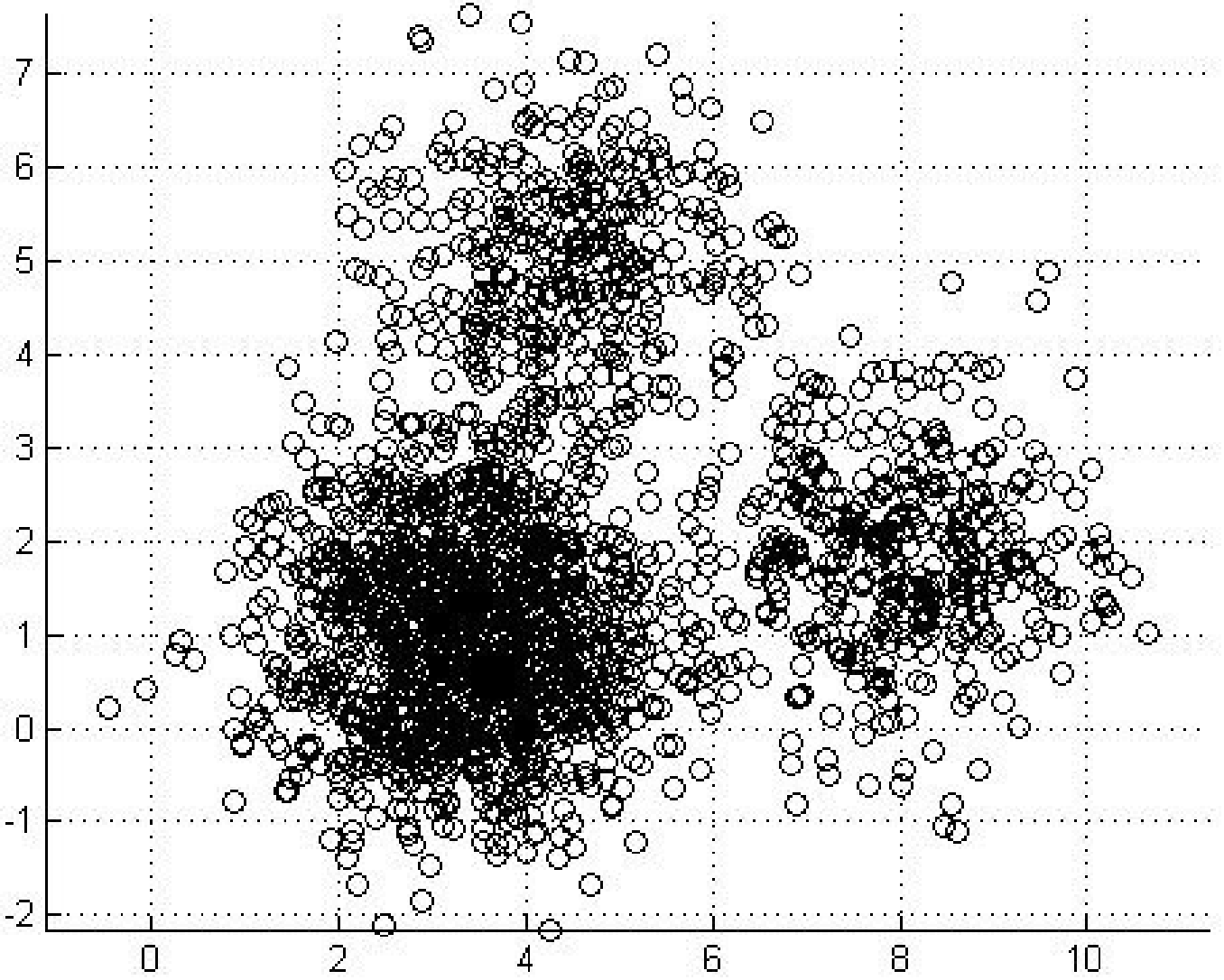
Seis clusters.

Cuatro clusters.

- La mejor definición de cluster depende de la naturaleza de los datos y los resultados deseados.

http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

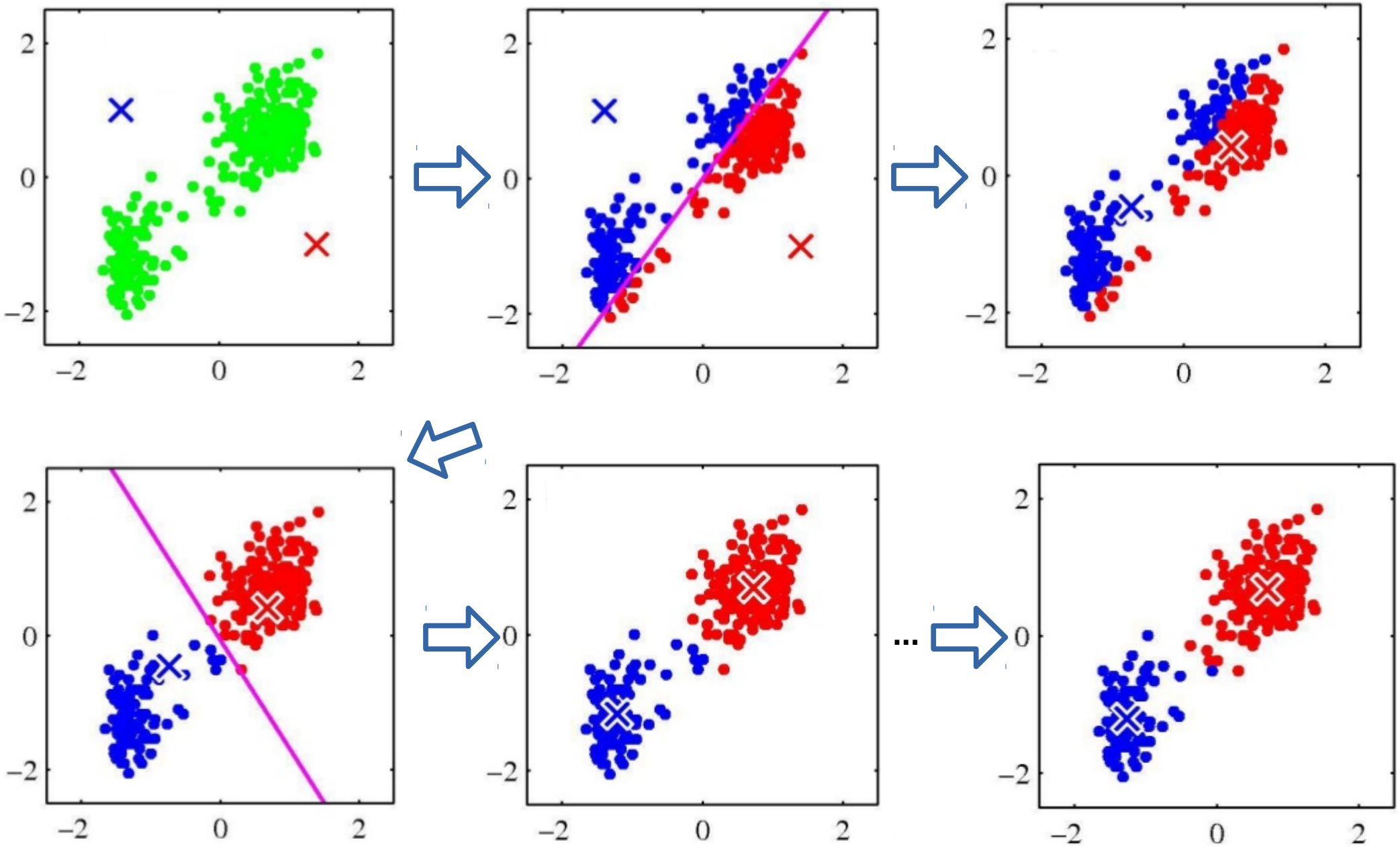
Clustering



Algoritmos de Clustering

1) *K*-Means Clustering
2) Clustering Jerárquico Aglomerativo
3) DBSCAN: Basado en densidades

K-Means Clustering



CM Bishop, Pattern Recognition and Machine Learning, Fig 9.1, p426

K-Means Clustering

- Datos: $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ instancias de D dimensiones.
- Objetivo: particionar datos en *K* clusters (para un *K* dado).
- Algoritmo:
  - Elegir *K* centroides al azar: $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$
  - Repetir:
    1) Asignar cada instancia al centroide más cercano.
    2) Recomputar el centroide de cada cluster.
  - Hasta que los centroides no cambien.
- Complejidad: (*N*=#instancias, *D*=#atributos, *I*=#iteraciones)
  - Tiempo: $O(I \cdot N \cdot K \cdot D)$
  - Espacio: $O((N + K) \cdot D)$

K-Means Clustering

- Definimos la distorsión del clustering de esta manera:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \boldsymbol{\mu}_k||^2$$

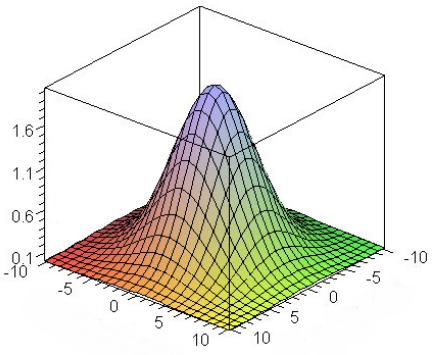
donde $r_{nk} = \begin{cases} 1 & \text{si } \mathbf{x}_n \text{ está asignada al cluster } k \\ 0 & \text{en caso contrario} \end{cases}$

- *J* es la suma de distancias^2 de cada instancia a su centroide.
- *K*-Means busca encontrar los $\{r_{nk}\}$ y $\{\boldsymbol{\mu}_k\}$ que minimicen *J*.
  - Paso 1) de *K*-means: minimiza *J* con respecto a los r_{nk}
  - Paso 2) de *K*-means: minimiza *J* con respecto a los $\boldsymbol{\mu}_k$
  - Ejemplo del Algoritmo EM.

K-Means Clustering

- Ventajas:
  - Simple, eficiente y general.
- Desventajas:
  - Hay que especificar *K*.
  - Sensible a ruido y outliers.
  - Muy sensible a la elección de los centroides iniciales.
    No siempre puede solucionarse con múltiples inicializaciones.
  - Sólo puede encontrar clusters globulares.

Mezclas de Gaussianas



- Distribución Normal o Gaussiana: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

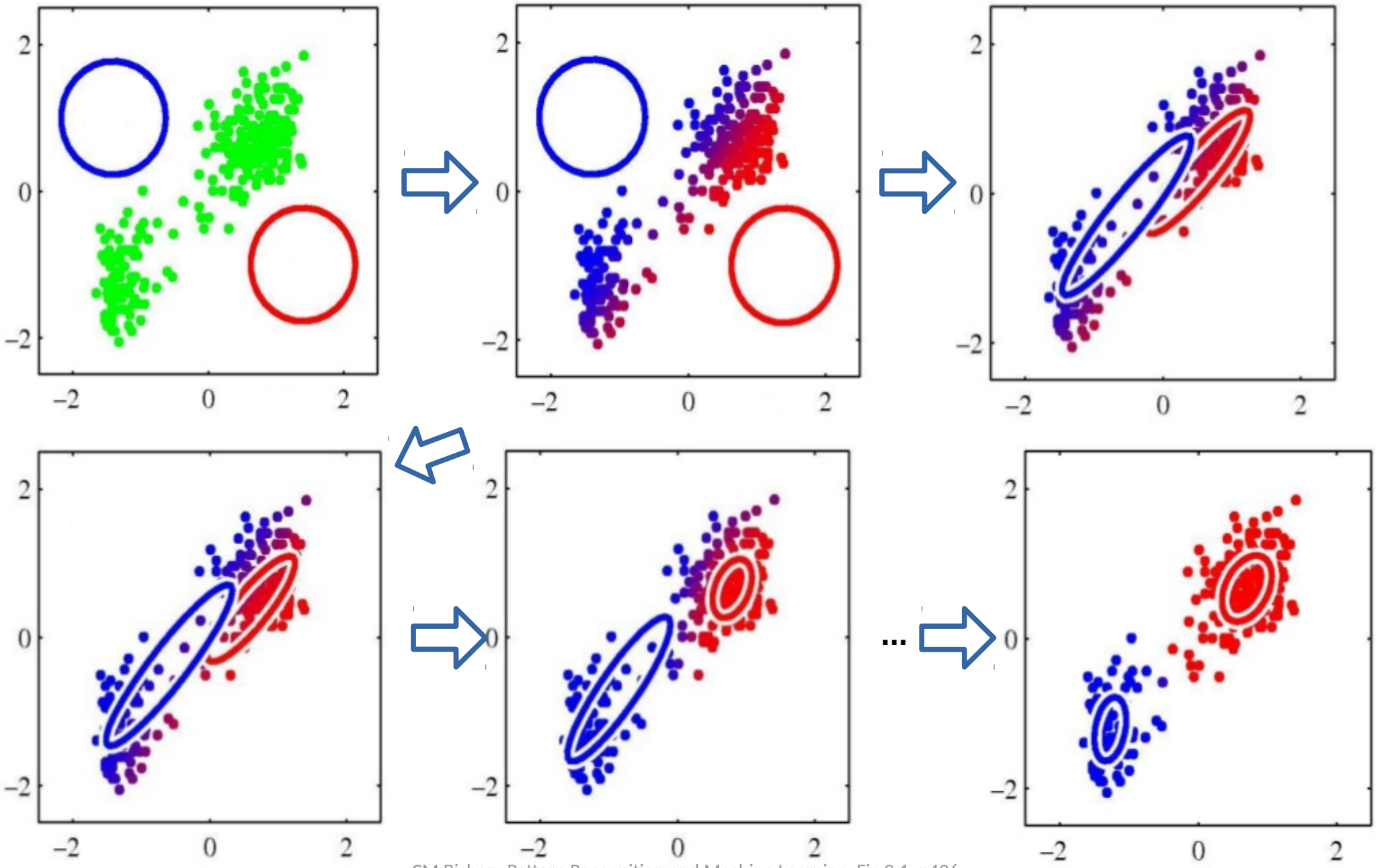
- Si podemos suponer que cada cluster sigue una distribución normal, entonces los datos se pueden ajustar a un modelo de mezclas de Gaussianas (GMM):

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- π_k son los coeficientes de mezcla, que cumplen:

$$0 \leq \pi_k \leq 1 \qquad \text{y} \qquad \sum_{k=1}^{K} \pi_k = 1$$

Mezclas de Gaussianas



CM Bishop, Pattern Recognition and Machine Learning, Fig 9.1, p426

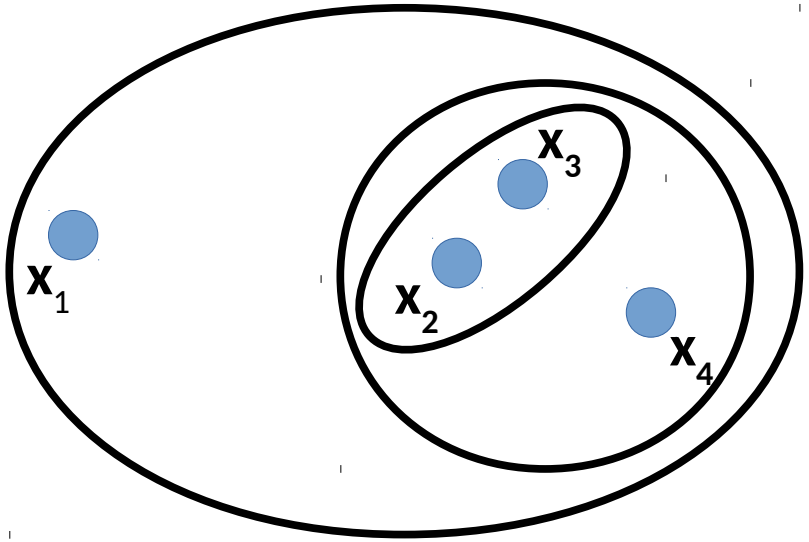
Mezclas de Gaussianas

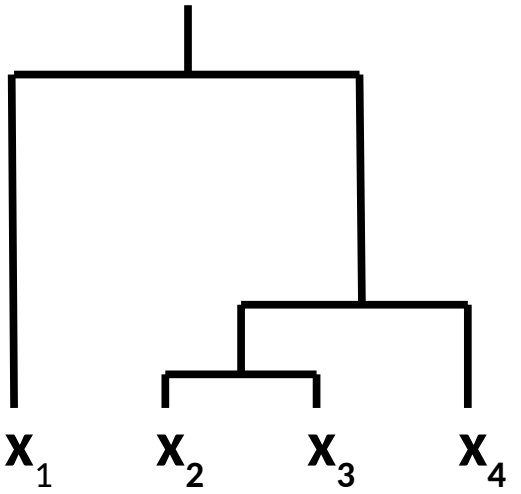
- Algoritmo EM para GMM:
  - Inicializar al azar: medias $\boldsymbol{\mu}_k$, covarianzas $\boldsymbol{\Sigma}_k$ y coeficientes π_k
  - Repetir:

    (E) Actualizar las pertenencias [0, 1] de cada instancia a las *K* componentes (clusters).

    (M) Actualizar el modelo: $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ y π_k
  - Hasta que las componentes no cambien.
- Aplicaciones de GMM:
  - Modelar la altura de las personas: diferentes grupos (ej: etarios, étnicos) siguen sus propias gaussianas.
  - Reconocimiento del habla: modelos acústicos de fonemas. Cada fonema (ej: /a/) puede producirse de diferentes formas (ej: dependiendo del contexto o del hablante).

Clustering Jerárquico Aglomerativo



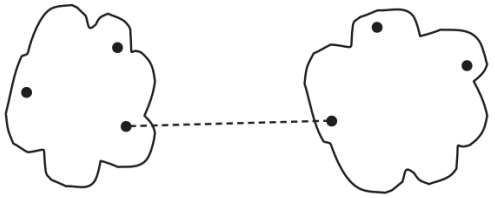


Dendrograma

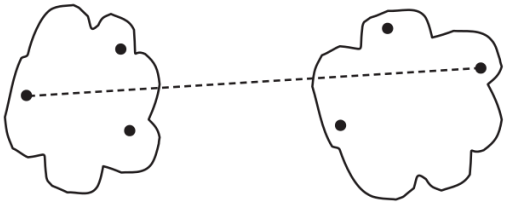
- Algoritmo:
  - Definir un cluster por cada instancia.
  - Repetir:
    - Fusionar los dos clusters más cercanos.
  - Hasta que quede un único cluster.
- Complejidad (N = #instancias):
  - Tiempo: $O(N^2 \log N)$
  - Espacio: $O(N^2)$

Clustering Jerárquico Aglomerativo

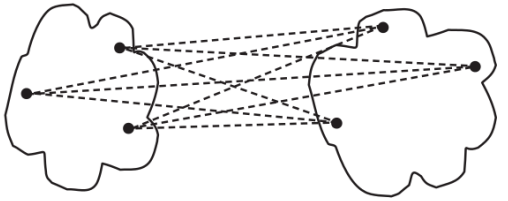
- ¿Cómo definir la distancia entre clusters?



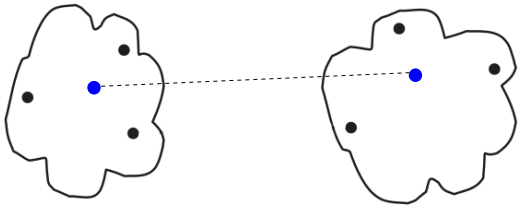
(a) MIN (single link.)



(b) MAX (complete link.)



(c) Group average.



(d) Cluster centroids.

...

  - Cada definición tiene sus pros y contras, respecto de la sensibilidad a ruido y outliers, y a la forma de los clusters que pueden manejar.

- El clustering jerárquico aglomerativo es bottom-up.
- También hay top-down: clustering jerárquico divisivo.

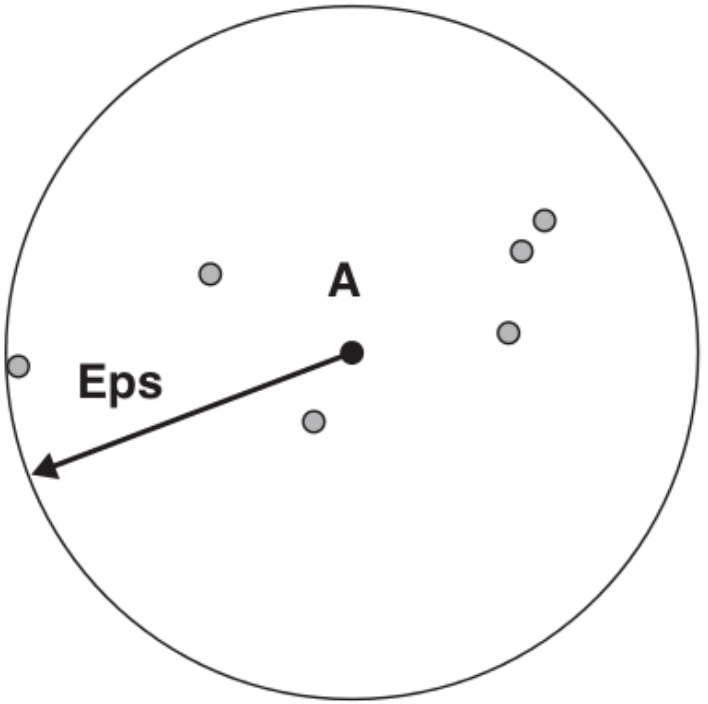
http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

Clustering Jerárquico Aglomerativo

- Ventajas:
  - No hay que especificar *K*.
  - Dendrograma: útil para crear taxonomías.
- Desventajas:
  - No busca optimizar una función objetivo global; toma sólo decisiones locales.
  - Caro computacionalmente.
  - Sensible a ruido y outliers.

DBSCAN

- *"**D**ensity-**b**ased **s**patial **c**lustering of **a**pplications with **n**oise"*
- Vecindad de un punto *A*: esfera centrada en *A* y radio *Eps*.
- Densidad de un punto *A*: cantidad de puntos dentro de la vecindad de A.

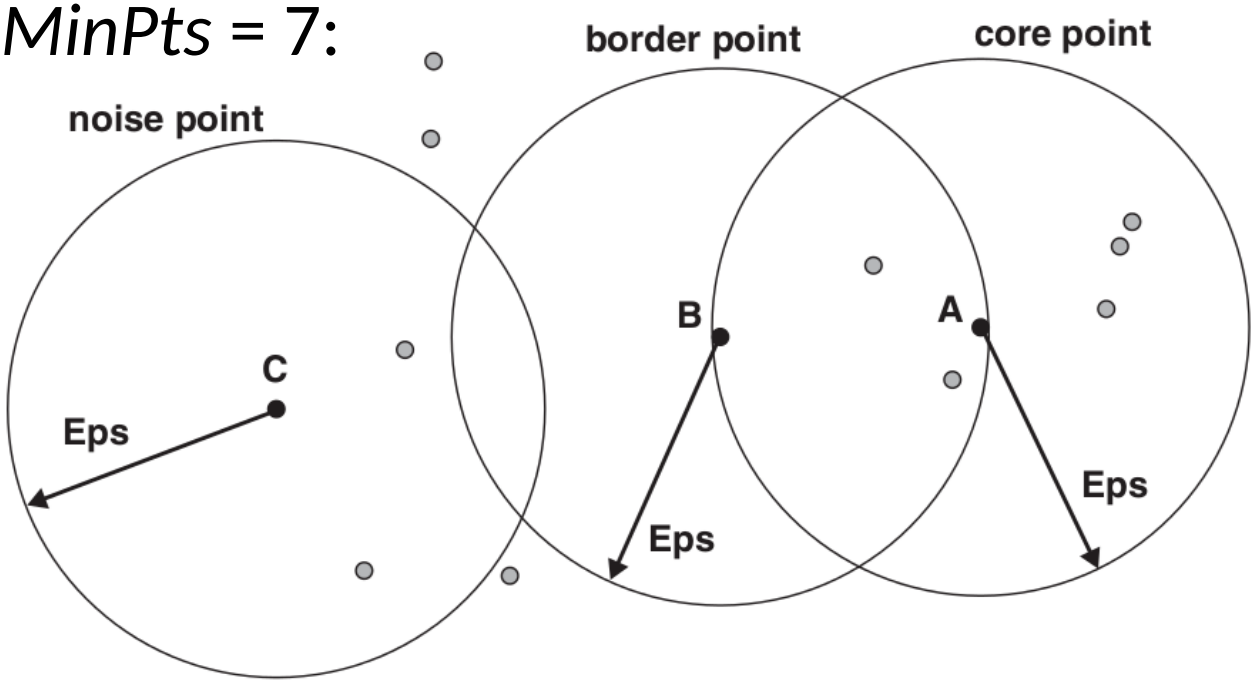


http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

DBSCAN

- *Core points*: Puntos con densidad mayor que la cte. *MinPts*.
- *Border points*: Puntos que no son *core*, pero son vecinos de algún punto *core*.
- *Noise points*: Puntos que no son *core* ni *border*.

Ejemplos, para *MinPts* = 7:



http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

DBSCAN

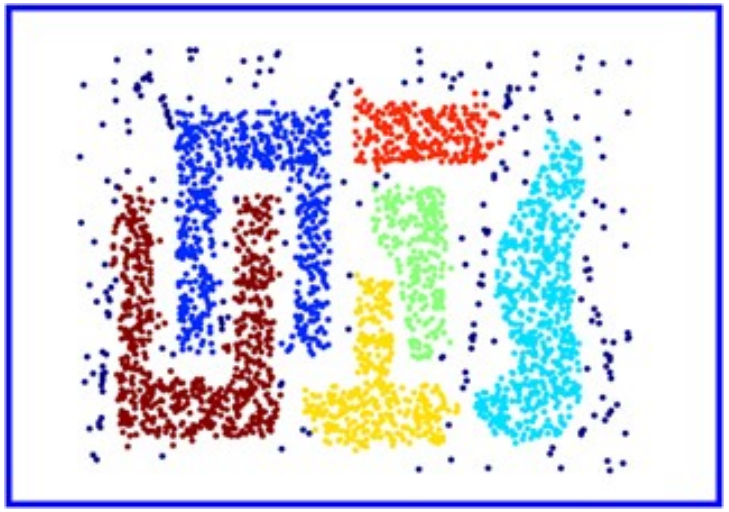
- Algoritmo:
  1) Etiquetar cada punto como *core*, *border* o *noise*.
  2) Eliminar todos los puntos *noise*.
  3) Poner una arista entre cada par de puntos *core* que son vecinos entre sí.
  4) Cada componente conexa corresponde a un cluster.
  5) Asignar los puntos *border* a uno de los clusters vecinos.

     Puede ser necesario desempatar entre 2+ clusters.
- Complejidad (N = #instancias):
  - Tiempo: O(N · tiempo de búsqueda en la vecindad)

    O(N^2) en el peor caso, O(N log N) si se puede usar *k-d trees*.
  - Espacio: O(N)

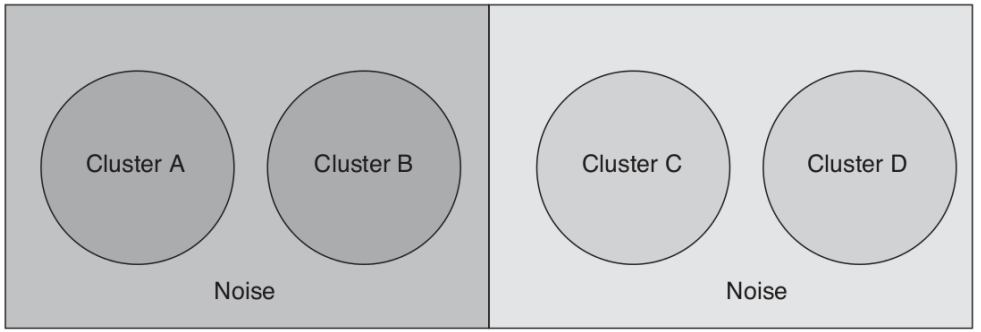
DBSCAN

- Ventajas:
  - No hay que especificar *K*.
  - Puede encontrar clusters de formas arbitrarias.
  - Es robusto al ruido.



http://www-users.cs.umn.edu/~kumar/dmbook/dmslides/chap8_basic_cluster_analysis.pdf

- Desventajas:
  - Elegir *Eps* y *MinPts* puede requerir tener conocimiento de los datos, y ser difícil en casos de alta dimensionalidad.
  - Funciona mal con datos con densidad variable:



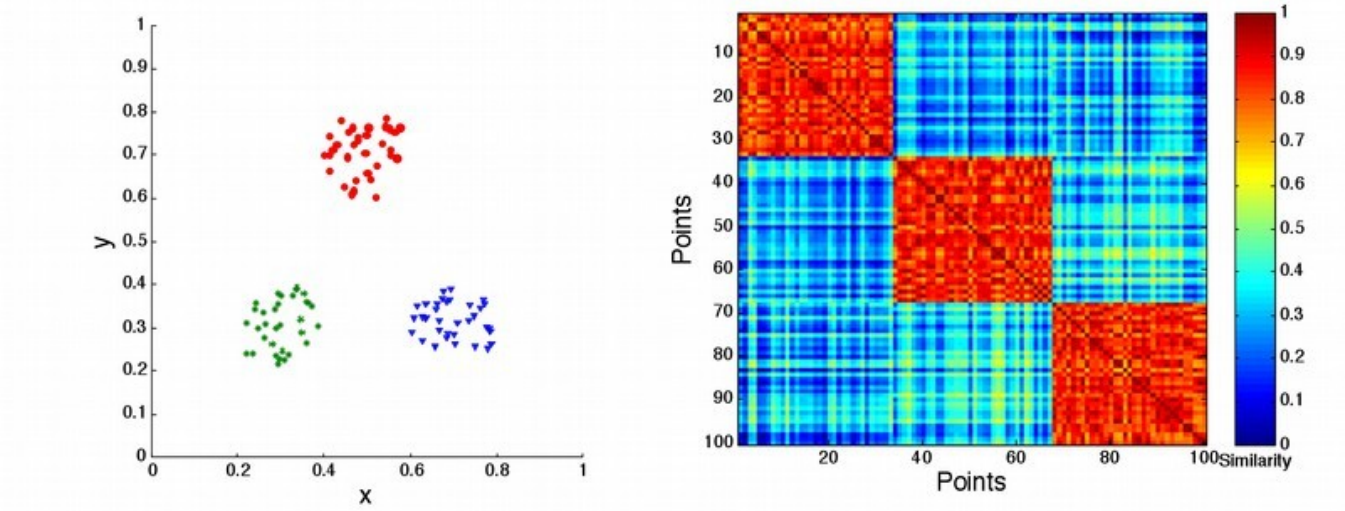
- Densidad indicada por el tono de gris.
- El ruido alrededor de A y B tienela misma densidad que C y D.
- DBSCAN colapsa A, B y el ruido que los rodea, o bien ignora C y D como si fueran ruido.

Evaluación de clusters

- Matriz de similitud (MS): matriz cuadrada y simétrica, con la similitud (valor [0, 1]) entre cada par de instancias.

	$\mathbf{x}_1$	$\mathbf{x}_2$	...	$\mathbf{x}_N$
$\mathbf{x}_1$	1	$\text{sim}(\mathbf{x}_1,\mathbf{x}_2)$	...	$\text{sim}(\mathbf{x}_1,\mathbf{x}_N)$
$\mathbf{x}_2$	$\text{sim}(\mathbf{x}_2,\mathbf{x}_1)$	1	...	$\text{sim}(\mathbf{x}_2,\mathbf{x}_N)$
...	...	...	...	...
$\mathbf{x}_N$	$\text{sim}(\mathbf{x}_N,\mathbf{x}_2)$	$\text{sim}(\mathbf{x}_N,\mathbf{x}_2)$	...	1

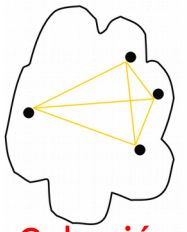
- Al ordenar las filas y columnas según el cluster de las instancias, el clustering es bueno si la MS es diagonal por bloques.



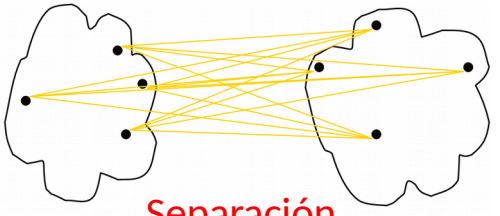
- Evaluación: inspección ocular de la MS; o cálculo de correlación entre la MS real y la MS ideal (diagonal por bloques).

Evaluación de clusters

- Métricas basadas en
  - cohesión: cuán estrechamente relacionados están los elementos de un mismo cluster;
  - separación: cuán distintos son los clusters entre sí.

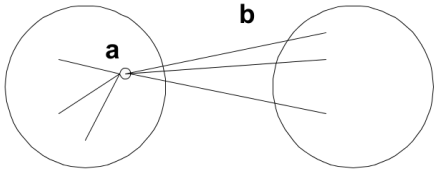


Cohesión



Separación

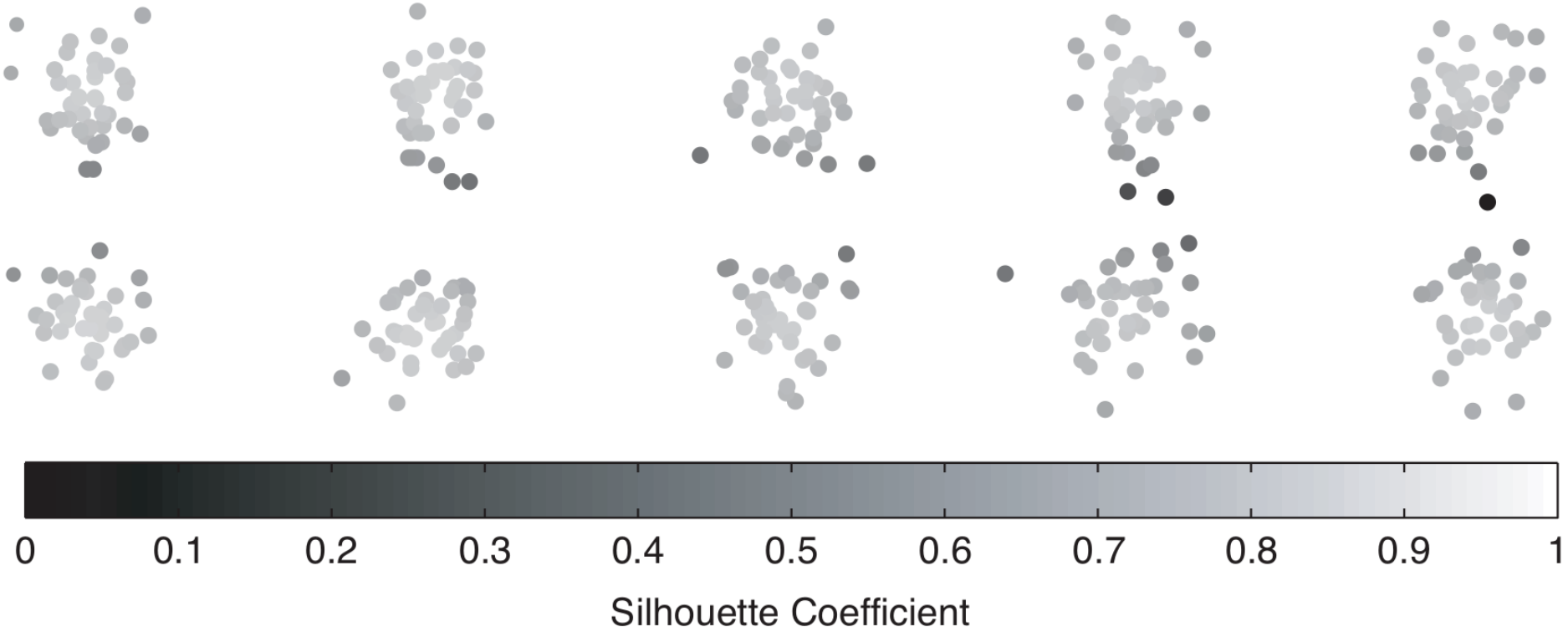
- Ejemplo: coeficiente *Silhouette*.
  - Para un punto $\mathbf{x}_i$:
    - $a \leftarrow$ distancia media de $\mathbf{x}_i$ a los puntos de su cluster
    - $b \leftarrow \text{mín}_{\text{C ε otros clusters}}$(distancia media de $\mathbf{x}_i$ a los puntos de C)
    - $s_i \leftarrow \begin{cases} 1 - a/b & \text{si } a<b \\ b/a - 1 & \text{si } a \geq b \quad \text{(inusual)} \end{cases}$



    - s_i se mueve en [−1, 1]. Cuanto más cerca de 1, mejor.

Evaluación de clusters

- Coeficiente *Silhouette.*



Resumen

- Algoritmos no supervisados: sin datos etiquetados.
- Técnicas de clustering:
  - *K*-Means. Mezclas de Gaussianas. Algoritmo EM.
  - Clustering Jerárquico.
  - BDSCAN (clustering basado en densidades).
- Evaluación de clusters:
  - Matriz de similitud.
  - Métricas basadas en cohesión y separación. Silhouette.