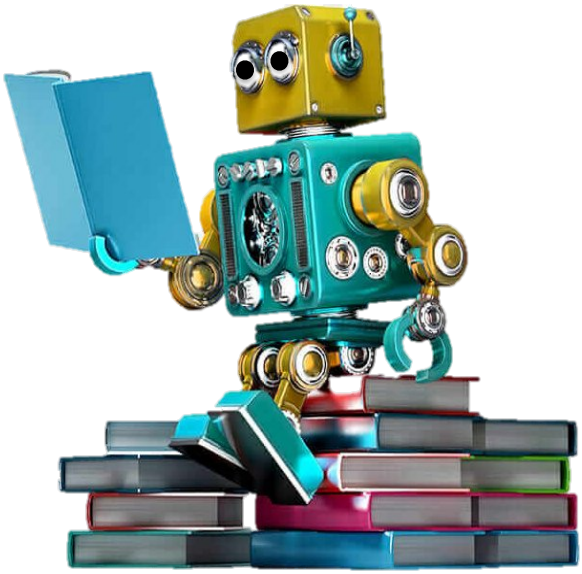
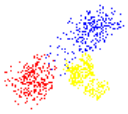
Clustering (Aprendizaje no supervisado)

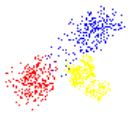


Quienes se parecen, se juntan. *(Refrán francés).*

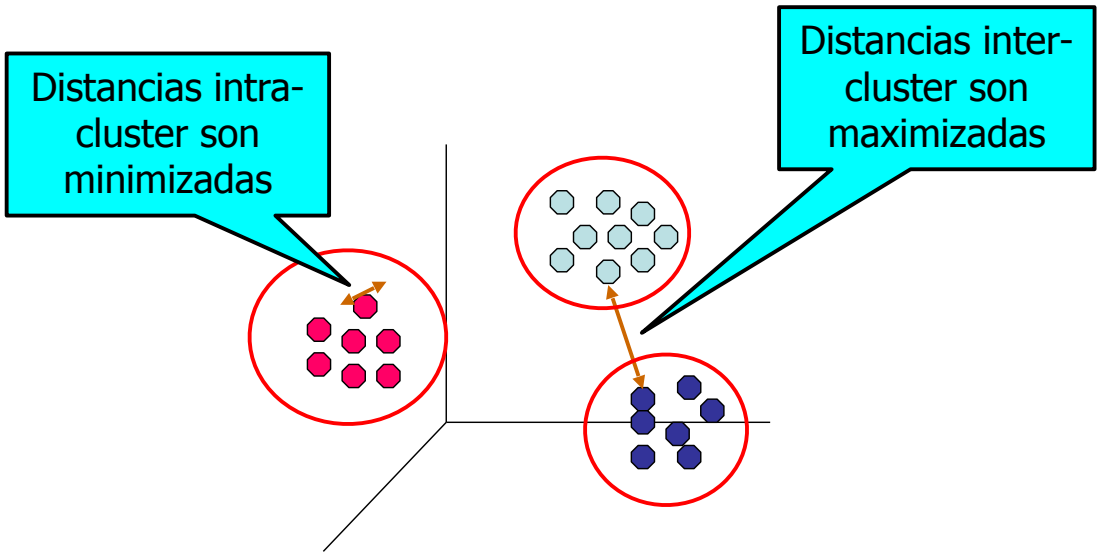


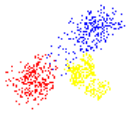
Clustering

- Técnica para aglomerar información en grupos (*clusters*) naturales.
- Constantemente distinguimos las cosas o las clasificamos en distintos grupos
- Ej: Distinguir entre perros y gatos, segmentar clientes, agrupar personas de distinto perfil, etc.
- La información no está rotulada (aprendizaje no supervisado)



Una idea de Clustering





¿Es fácil hacer clustering?

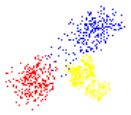
¿Cuántos clusters?

6

2

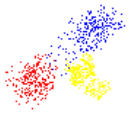
4

AMBIGUEDAD



Clustering

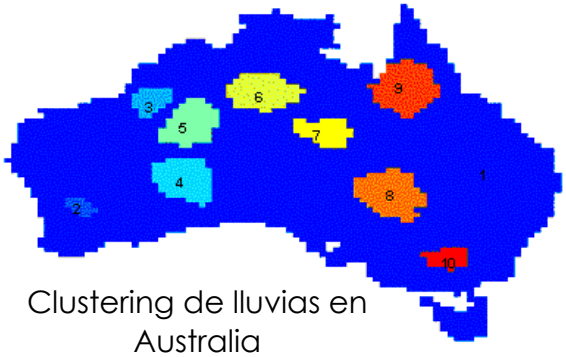
- Aplicaciones:
  - Agrupar clientes según distintos patrones de compra
  - Identificar Genes con funcionalidades similares
  - Identificar terrenos de características similares utilizando observaciones de la tierra
  - Identificar casas en una ciudad según ubicación, etc.
  - Sumarización de regiones



Aplicaciones

1. Imágenes.

Ejm:
Segmentación de terrenos

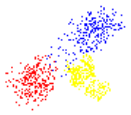


Clustering de lluvias en Australia

2. Textos.

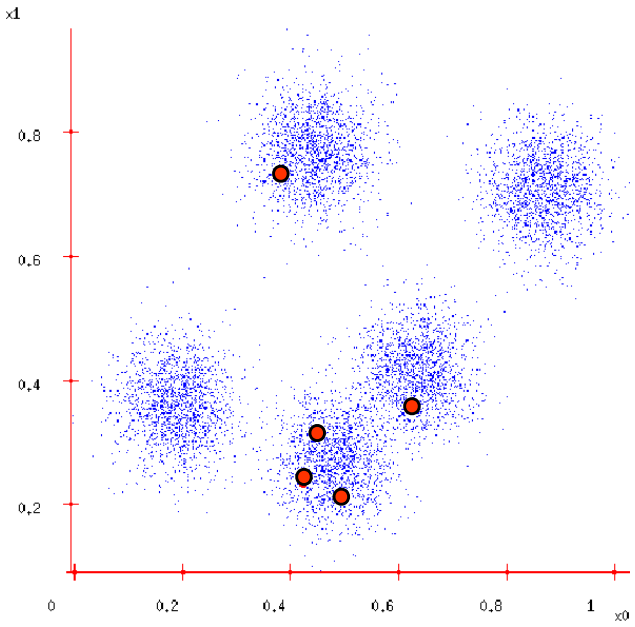
Ejm:
Documentos relacionados en relación a grupos de empresas

	Clusters descubiertos	Grupo
1	Applied-Matl-DOWN,Bay-Network-Down,3-COM-DOWN, Cabletron-Sys-DOWN,CISCO-DOWN,HP-DOWN, DSC-Comm-DOWN,INTEL-DOWN,LSI-Logic-DOWN, Micron-Tech-DOWN,Texas-Inst-Down,Tellabs-Inc-Down, Natl-Semiconduct-DOWN,Oracl-DOWN,SGI-DOWN, Sun-DOWN	Technology1-DOWN
2	Apple-Comp-DOWN,Autodesk-DOWN,DEC-DOWN, ADV-Micro-Device-DOWN,Andrew-Corp-DOWN, Computer-Assoc-DOWN,Circuit-City-DOWN, Compaq-DOWN, EMC-Corp-DOWN, Gen-Inst-DOWN, Motorola-DOWN,Microsoft-DOWN,Scientific-Atl-DOWN	Technology2-DOWN
3	Fannie-Mae-DOWN,Fed-Home-Loan-DOWN, MBNA-Corp-DOWN,Morgan-Stanley-DOWN	Financial-DOWN
4	Baker-Hughes-UP,Dresser-Inds-UP,Halliburton-HLD-UP, Louisiana-Land-UP,Phillips-Petro-UP,Unocal-UP, Schlumberger-UP	Oil-UP



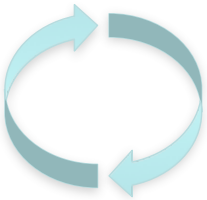
K-Means

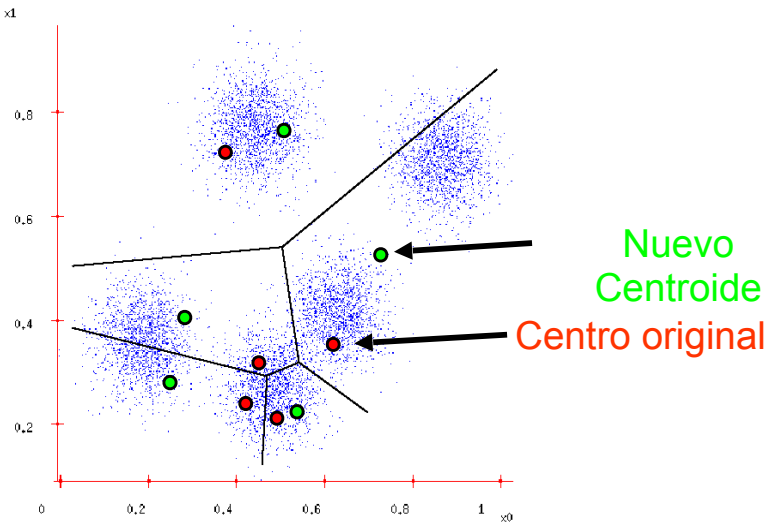
- "Adivinar" cuántos clusters son
- Dar centros iniciales aleatoriamente

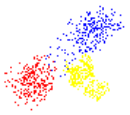


K-Means

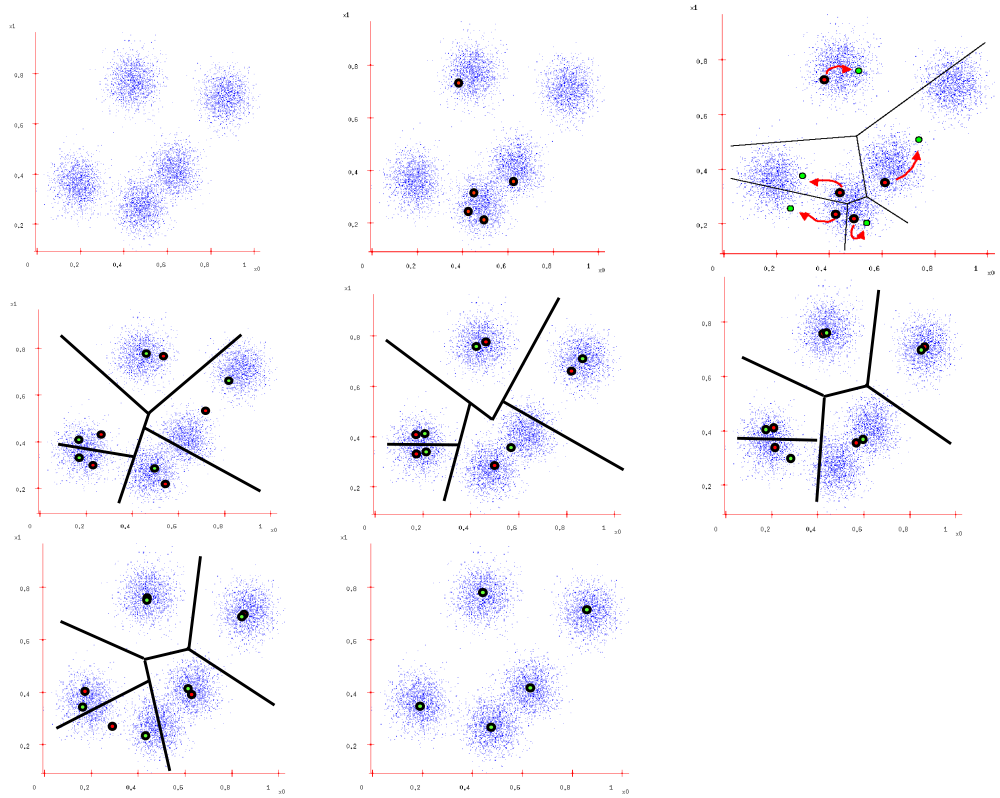
- Asignar a cada punto el centro más cercano
- Cada centro calcula el centroide de los puntos que fueron asignados a él
- Los centros se desplazan a dichos centroides

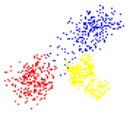






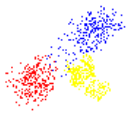
K-Means





K-Means

1. Inicializa K centros en forma aleatoria
2. Asignar cada punto en el set de datos al centro más cercano
3. Re-estimar la posición de los centros calculado el valor medio de los puntos que le fueron asignados
4. Repetir pasos 2 y 3 hasta que la posición de los centros no cambie en forma significativa entre iteraciones sucesivas



K-Means

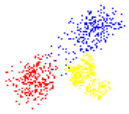
Problema: No siempre se converge a la posición óptima de los centros.

Idea: Correr K-Means varias veces partiendo de distintos puntos de partida.

Tiempo de ejecución de K-Means para n datos, d variables y k clusters

- K-Means $O(ndk)$
- Solución óptima: $O(n^{dk+1} \log n)$

Halla los K medias, pero **no** es el K-Means visto.



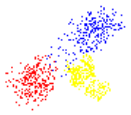
Mean Shift

- Idea: Los centros de los clusters se ubican en sectores de mayor densidad de datos
- Este algoritmo considera una vecindad local a cada centro y mueve el centro en la dirección de mayor aumento de densidad

- Su complejidad es aproximadamente , n datos y d dimensiones:

$$O(n^2 d)$$

Usualmente mas lento que K-means



Pseudocodigo de Mean Shift

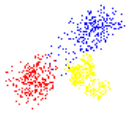
- Inicializar K medias $x_i, \quad i = 1:k$
- Calcular nueva media con (*h: bandwith*):

$$m_h(x) = \frac{\sum_{i=1}^{n} x_i g\left(\left\| \frac{x - x_i}{h} \right\|\right)}{\sum_{i=1}^{n} g\left(\left\| \frac{x - x_i}{h} \right\|\right)}$$

Kernel Gaussiano
(1 ejm de muchos)

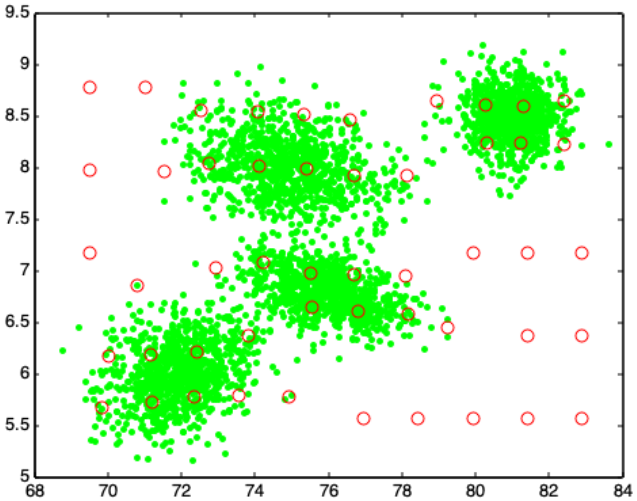
$$g\left(\left\| \frac{x - x_i}{h} \right\|\right) \equiv g(x_i; x, h) = e^{\frac{-|x - x_i|^2}{h^2}}$$

- Actualizar: $x_i \Rightarrow m_h(x_i)$
- Iterar hasta converger (no cambio en x)

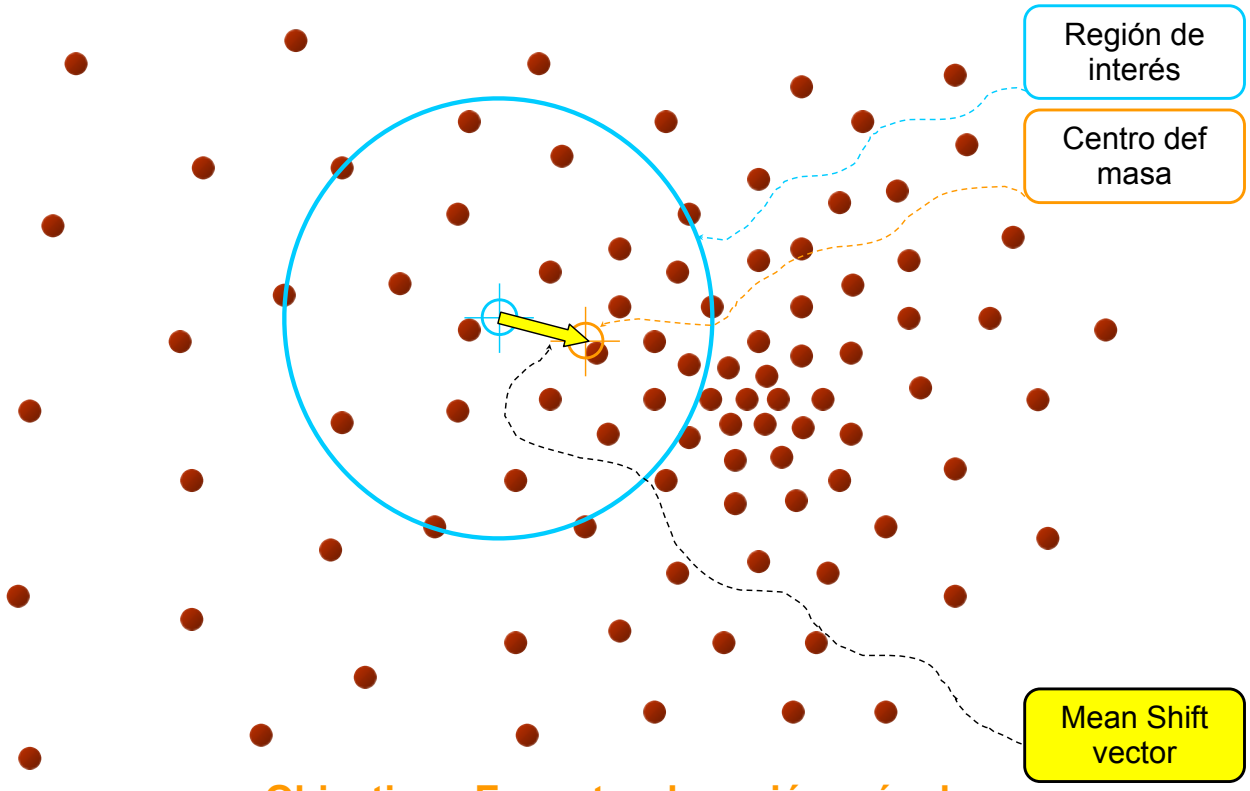


Mean Shift

1. Inicio
   - Especificar el tamaño de la ventana
   - Especificar una gran cantidad de centros de manera de cubrir el espacio de hipótesis

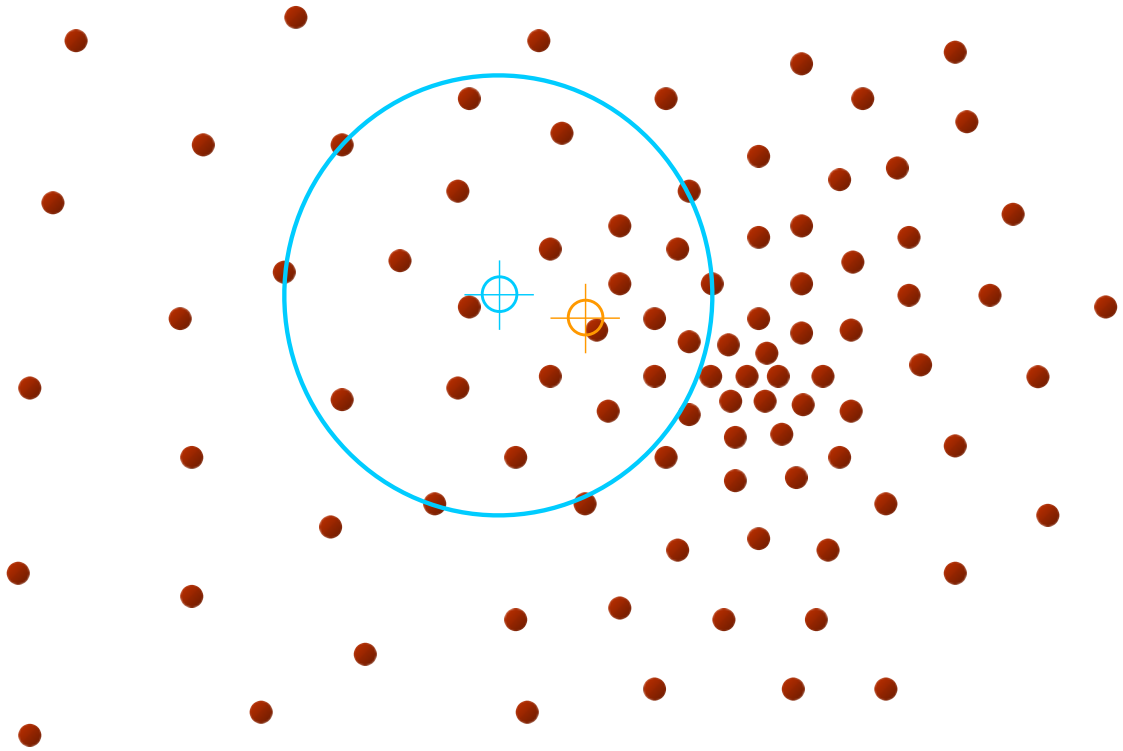


Mean Shift

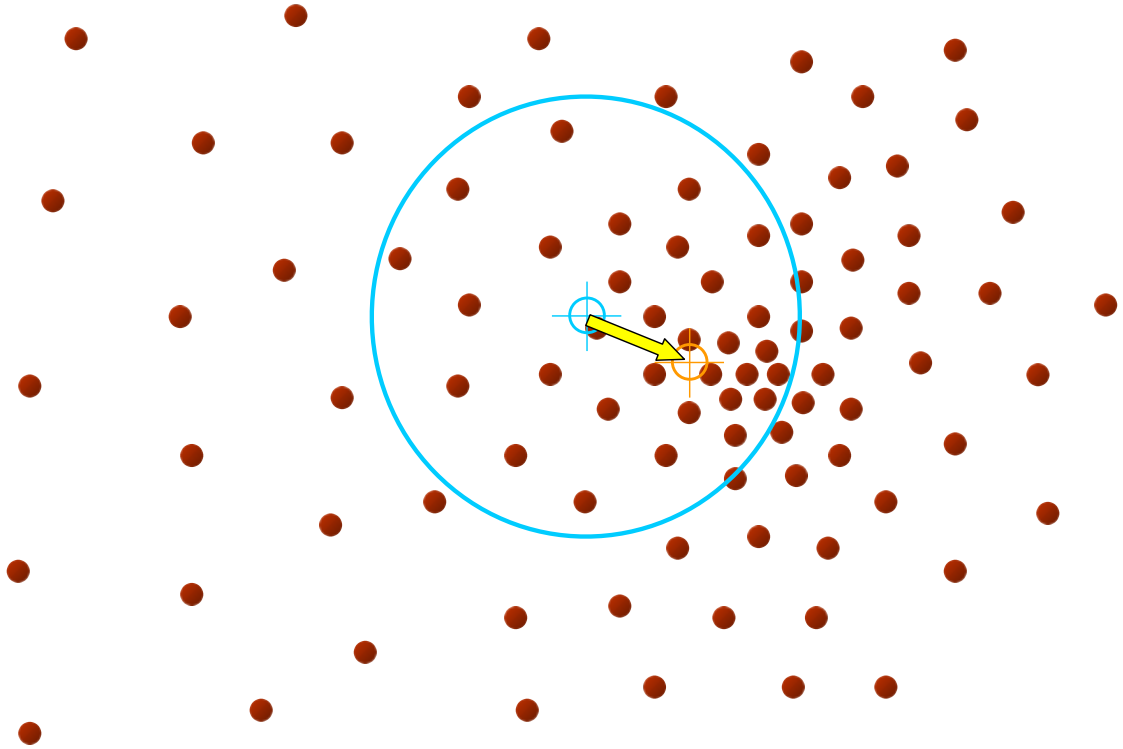


Objectivo : Encontrar la región más densa

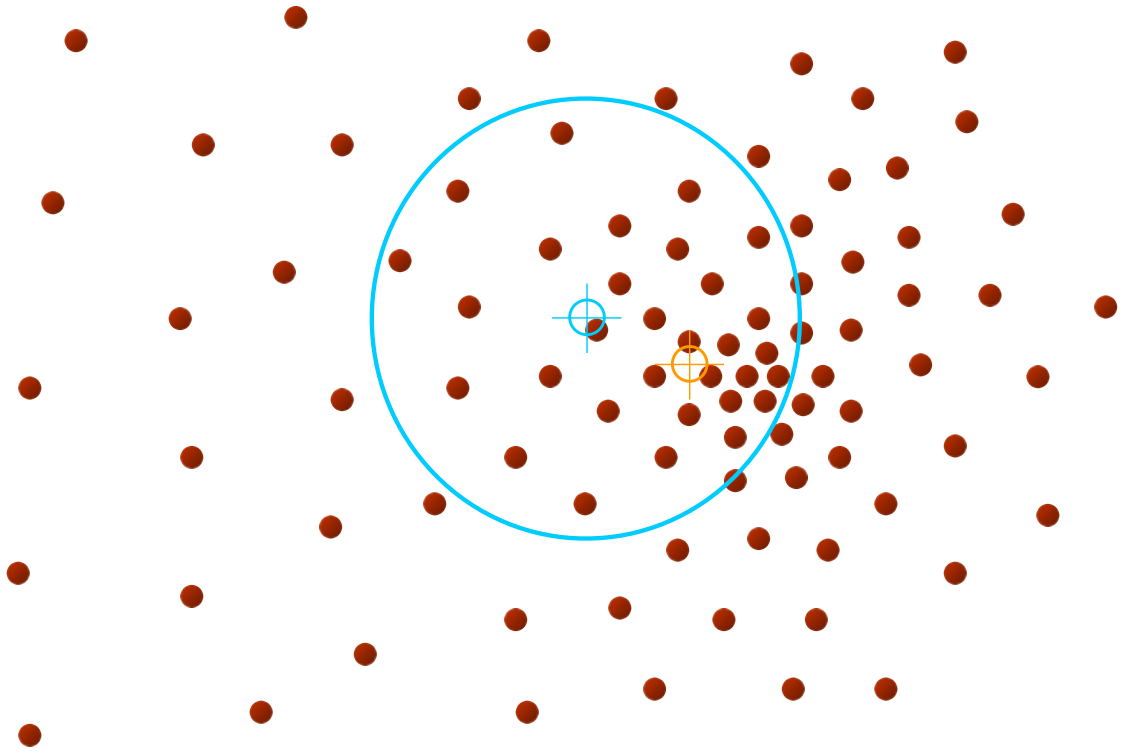
Mean Shift



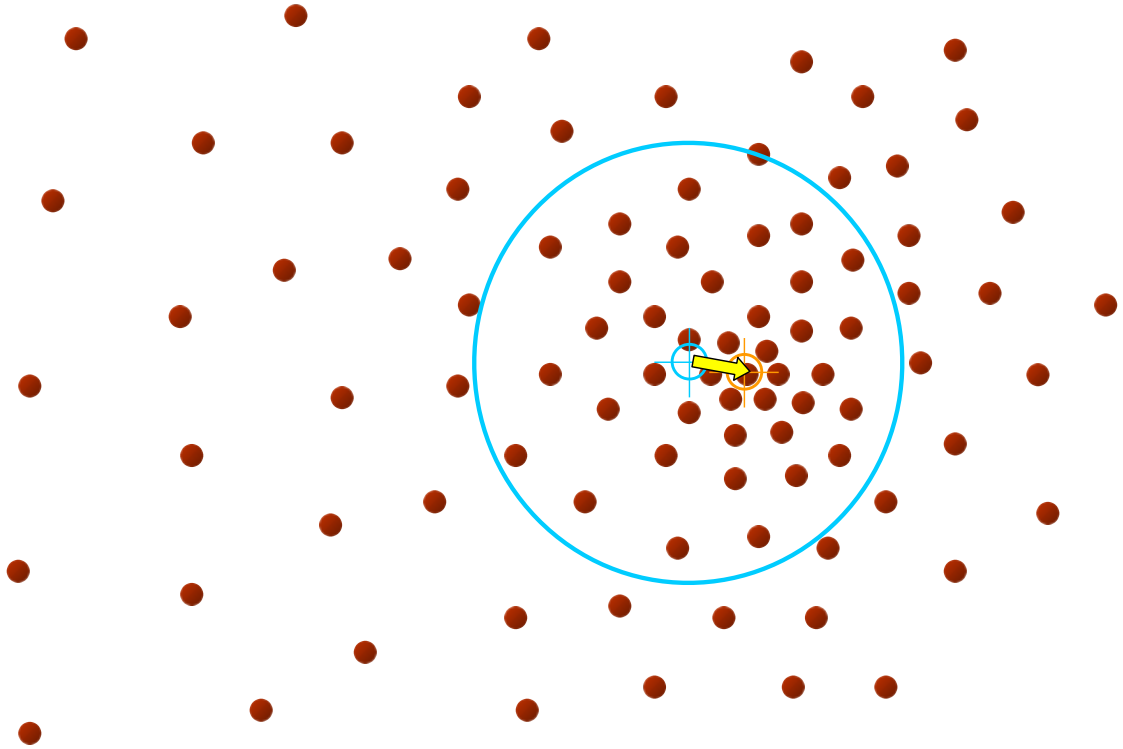
Mean Shift



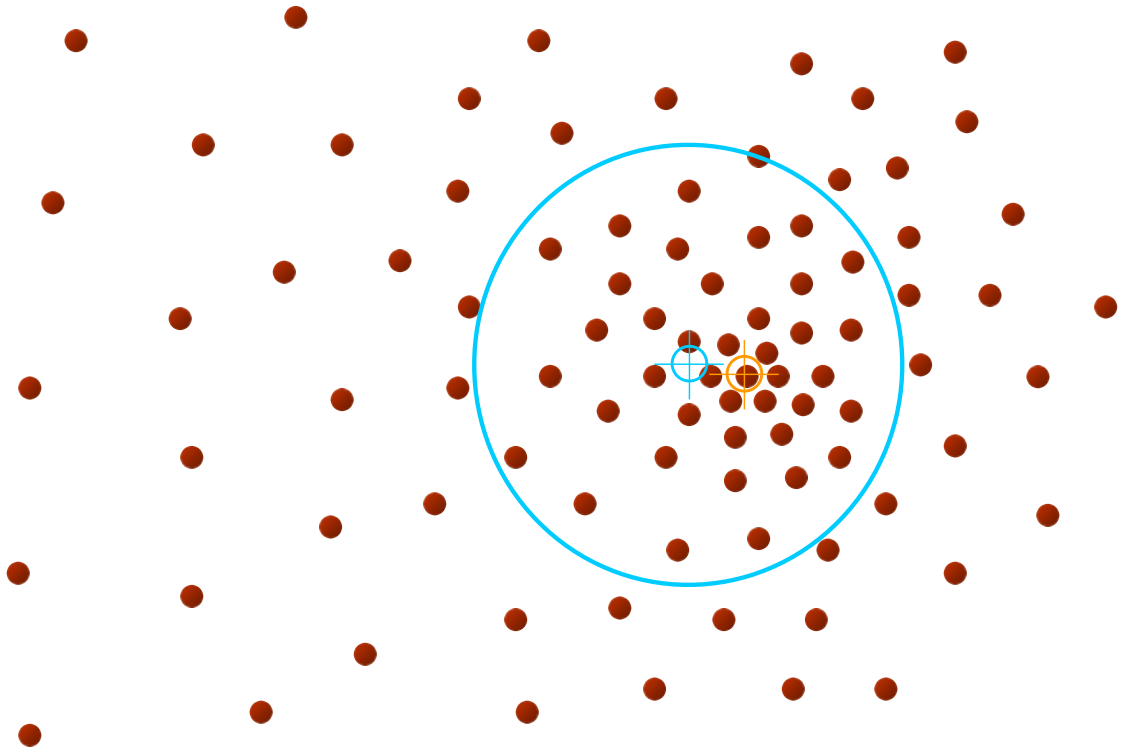
Mean Shift



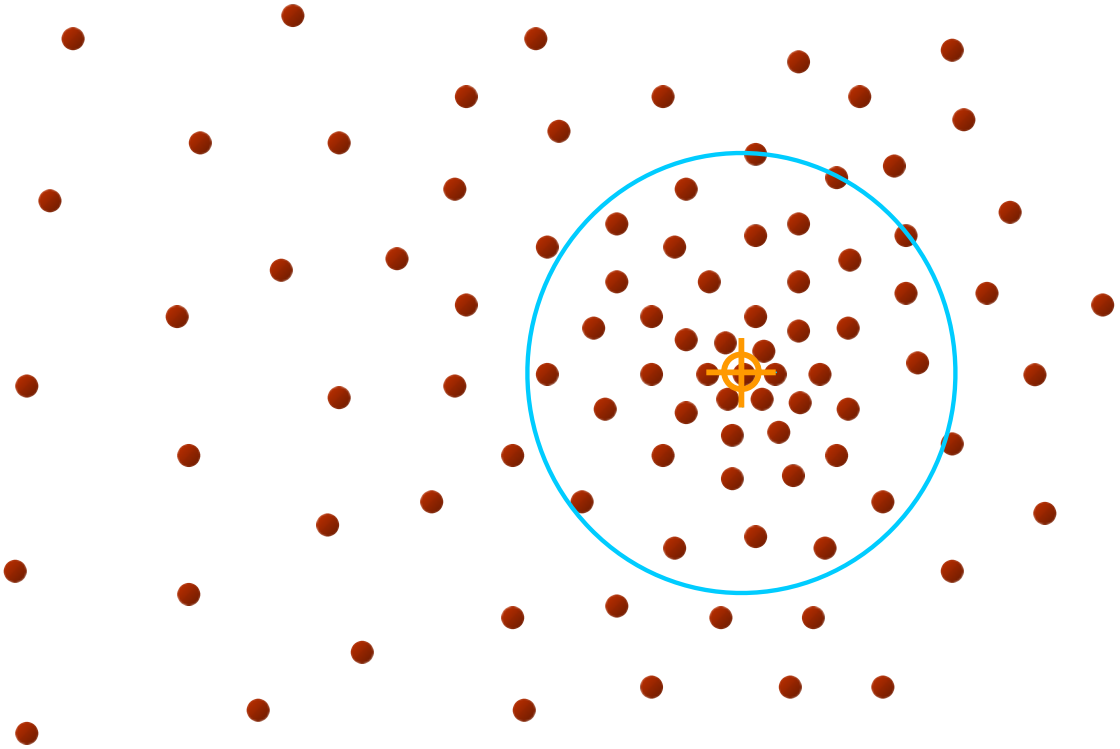
Mean Shift

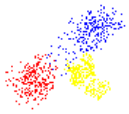


Mean Shift



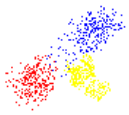
Mean Shift





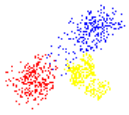
Revisando mas: taxonomía

- Los algoritmos de clustering se clasifican principalmente como:
  - Particionales, jerárquicos y por densidad.
- Los algoritmos que hemos visto hasta ahora caen en la categoría particional



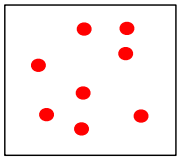
Clustering Jerárquico

- Los algoritmos de clustering jerárquico aglomeran o dividen puntos guiados por alguna métrica de similaridad,
- De allí que hay 2 tipos:
  - Aglomerativos
  - Divisivos

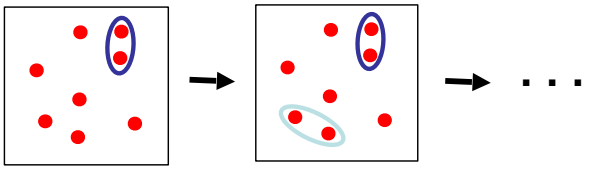


Clustering Aglomerativo

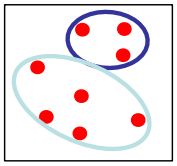
- Inicio: cada punto es un cluster

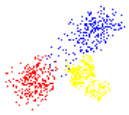


- Iterativamente unir los dos clusters más cercanos

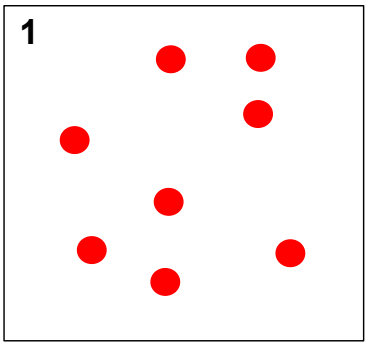


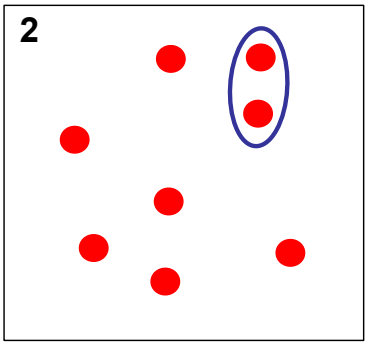
- Parar cuando distancia entre los clusters a unir supera algún umbral pre-determinado

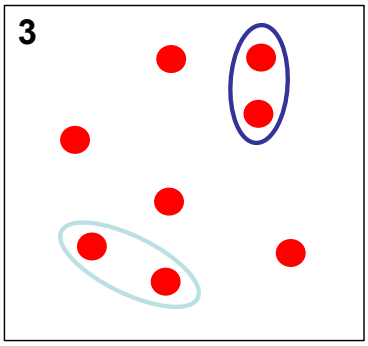


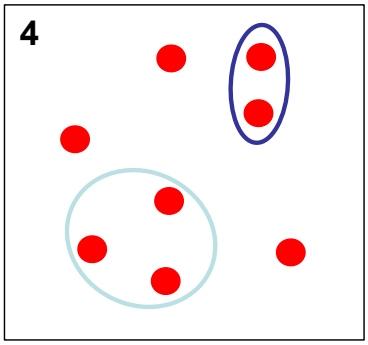


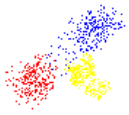
Clustering Aglomerativo



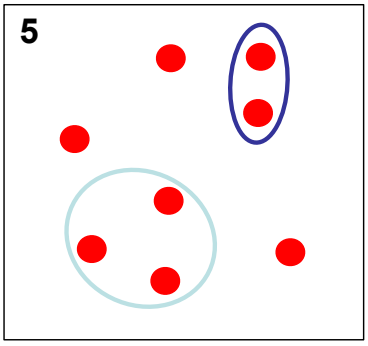


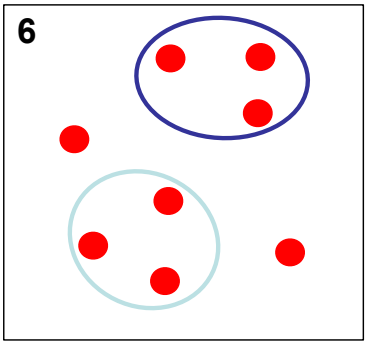


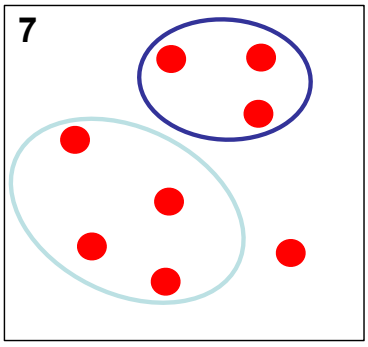


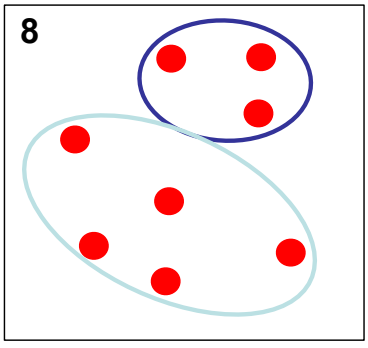


Clustering Aglomerativo







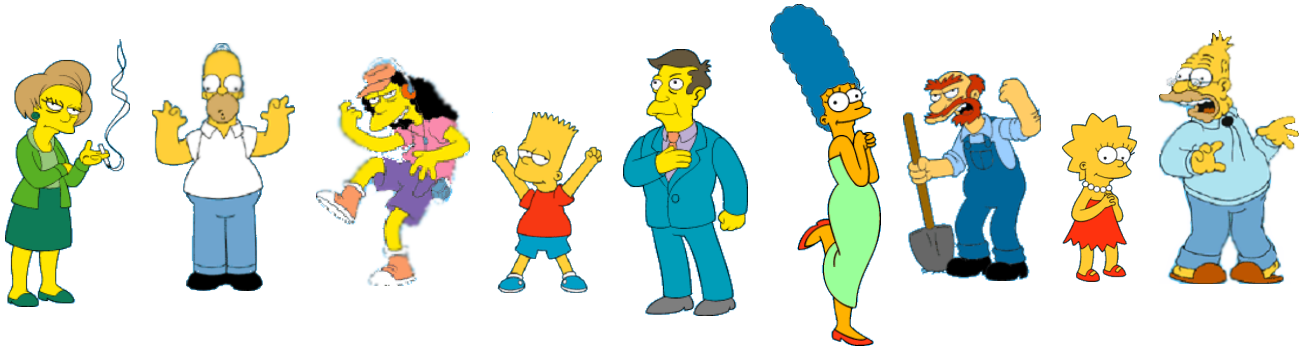


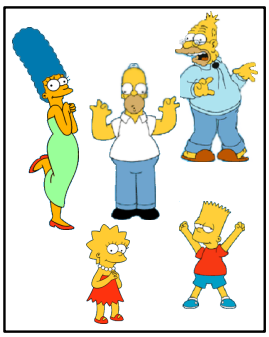
Clustering Aglomerativo

- Este método depende de el objetivo ya que se debe definir un criterio de detención del algoritmo, sino se llega a un cluster que contiene a todos los elementos.
- El resultado dependerá fuertemente de las medidas de similaridad

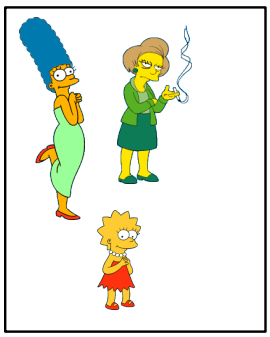


Medidas de similaridad

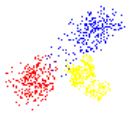



La familia Simpson


Trabajadores de la escuela


Mujeres


Hombres



Distancia entre clusters

4 métodos más comunes:

Conexión simple (single link)

$$D(C_a, C_b) = Min\{d(i,j)\}, i \in C_a, j \in C_b$$

Conexión completa (complete link)

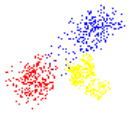
$$D(C_a, C_b) = Max\{d(i,j)\}, i \in C_a, j \in C_b$$

Distancia entre medias (mean distance)

$$D(C_a, C_b) = D(\mu_a, \mu_b)$$

Distancia promedio entre pares (av. pairwise dist.)

$$D(C_a, C_b) = avg\{d(i,j)\}, i \in C_a j \in C_b$$

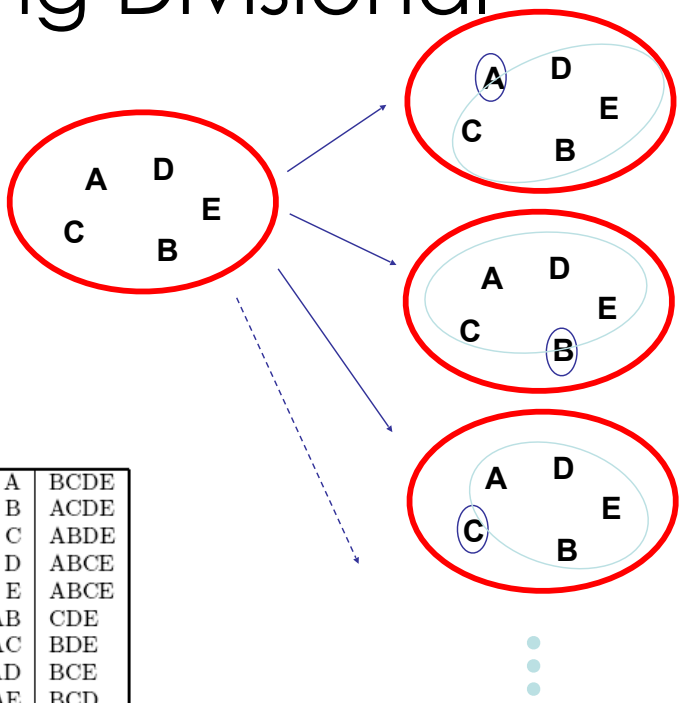


Clustering Divisional

- Mientras el cluster aglomerativo es bottom-up, el cluster divisional es top-down
- Al comienzo todos los puntos son un cluster
- Este mega-cluster es luego dividido. De preferencia usando algoritmo particional.
- Se sigue iterando, hasta que cada punto sea su propio cluster.

Clustering Divisional

A	B	C	D	E	
0	1	2	2	3	A
	0	2	4	3	B
		0	1	5	C
			0	3	D
				0	E

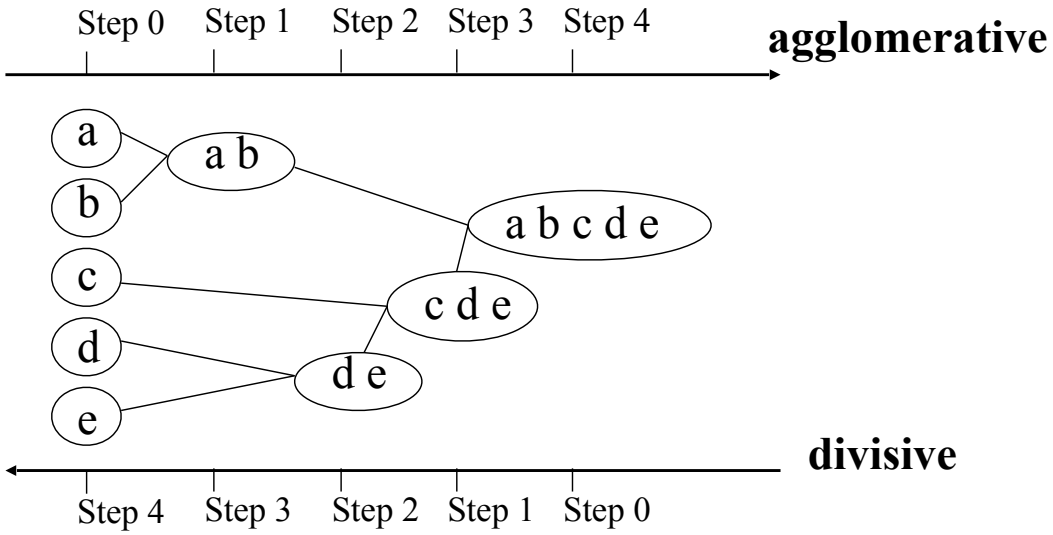


Posibles splits

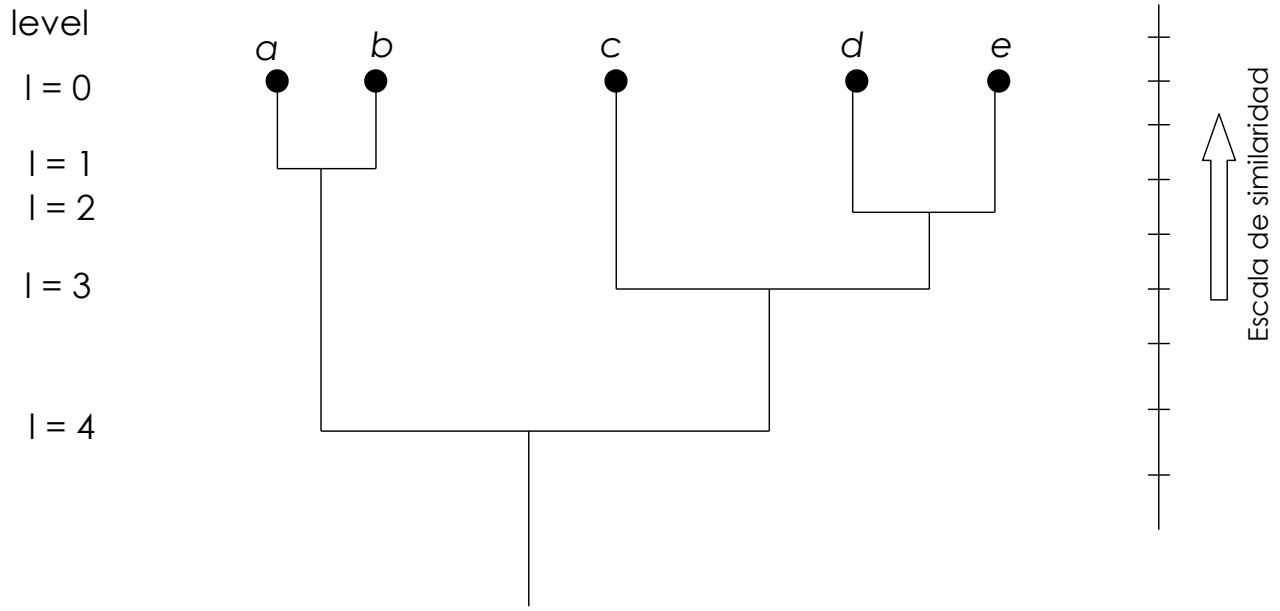
A	BCDE
B	ACDE
C	ABDE
D	ABCE
E	ABCE
AB	CDE
AC	BDE
AD	BCE
AE	BCD
BC	ADE
BD	ACE
BE	ACD
CD	ABE
CE	ABD
DE	ABC

Complejidad muy alta,
Por eso algoritmo k-Means

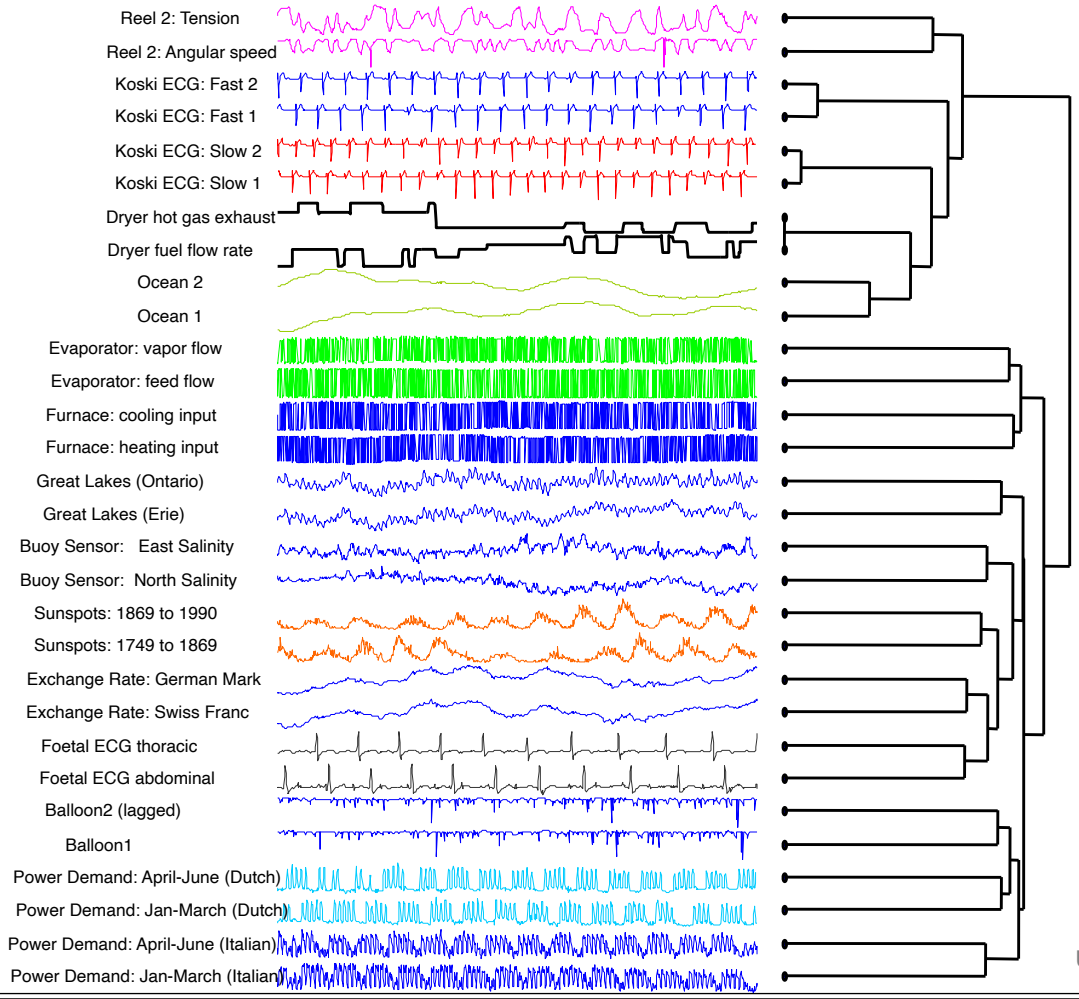
Clustering Divisional

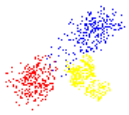


Dendograma

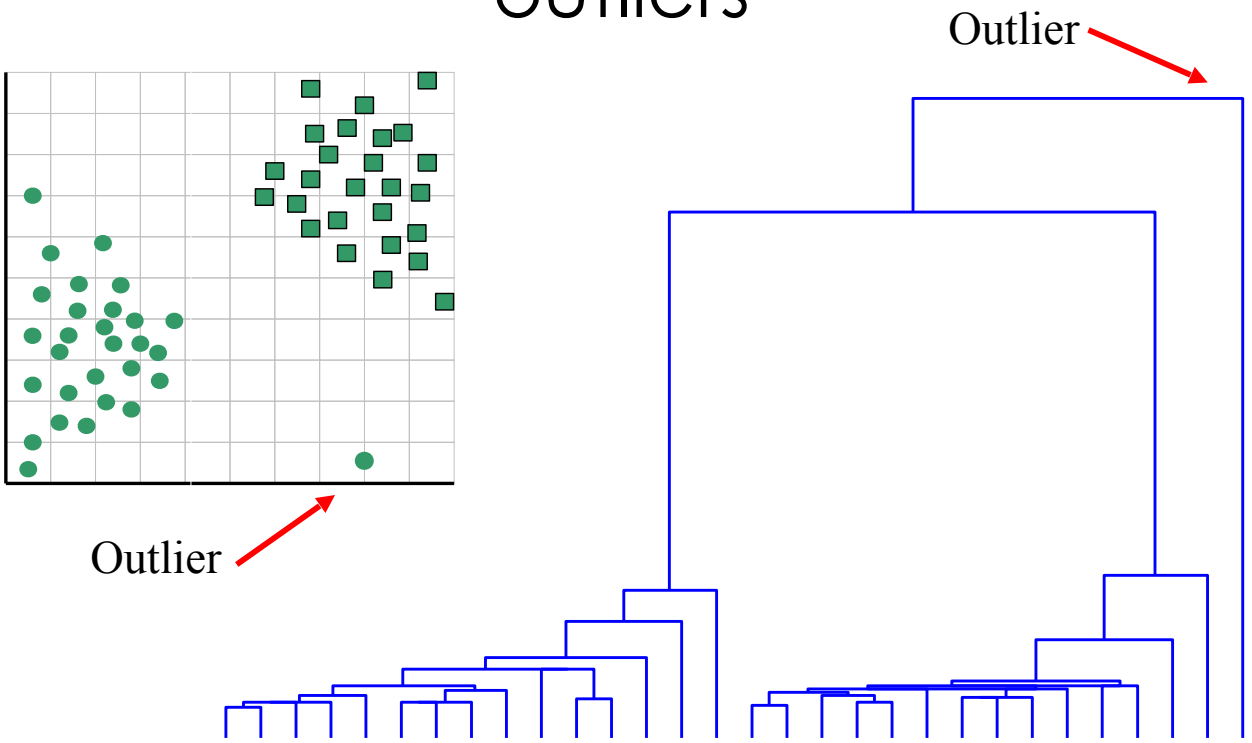


Estructura de árbol utilizada para representar el proceso de clustering jerárquico



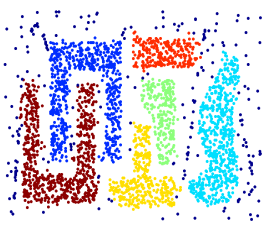


Dendograma y detección de outliers



Clustering basado en densidad

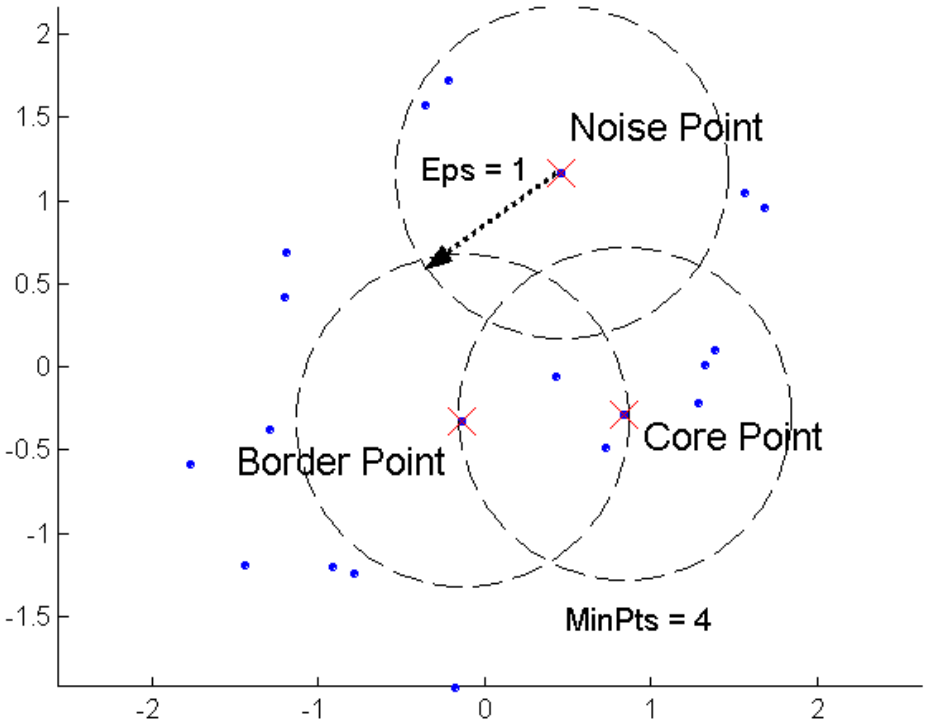
- Estos algoritmos se basan en unir datos de acuerdo a cercanía y densidad local.
- Los algoritmos mas conocidos son:
  - DBScan
  - Optics
  - Chameleon
  - Aplicados en cluste

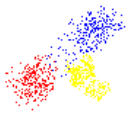


DBSCAN (clustering por densidad)

- Algoritmo de clustering que explora regiones de densidades altas formando clusters de formas arbitrarias
- Algunas definiciones:
  - Un objeto es **core point** si dentro de una vecindad de radio ***Eps*** contiene al menos una cantidad dada de puntos (***MinPts***).
  - Un objeto es **border point** si tiene menos de ***MinPts*** puntos dentro de radio **Eps** pero es vecino de un core point**.**
  - Un objeto es **noise point** si no es border ni core point.

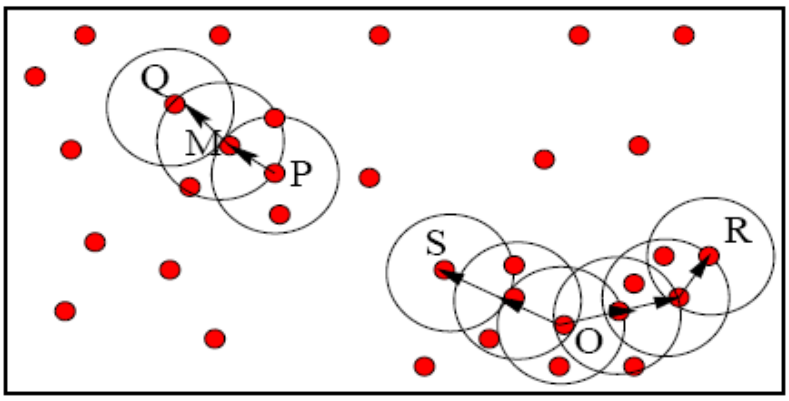
DBSCAN



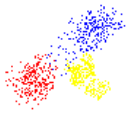


DBSCAN

- Ej

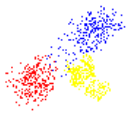


- Supongamos que MinPts = 2 y la vecindad está dada por el radio de los círculos de la figura.
- M,P,O y R son "core points"
- Q es directamente alcanzable por densidad desde M.
- M es directamente alcanzable por densidad desde P y vice versa



DBSCAN

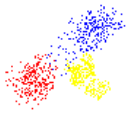
- Q es (indirectamente) alcanzable por densidad desde P porque Q es directamente alcanzable por densidad desde M y M es directamente alcanzable por densidad desde P.
- P no es alcanzable por densidad desde Q porque Q no es un "core object"
- R y S son alcanzables por densidad desde O
- O es alcanzable por densidad desde R
- O,R y S están conectados por densidad



Algoritmo de DBSCAN

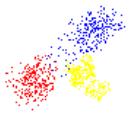
- Encontramos los core, border y noise points.
- Eliminamos los noise points.

$current\_cluster\_label \leftarrow 1$
for all core points **do**
  if the core point has no cluster label **then**
    $current\_cluster\_label \leftarrow current\_cluster\_label + 1$
    Label the current core point with cluster label $current\_cluster\_label$
  end if
  for all points in the Eps-neighborhood, except i^{th} the point itself **do**
    if the point does not have a cluster label **then**
      Label the point with cluster label $current\_cluster\_label$
    end if
  end for
end for



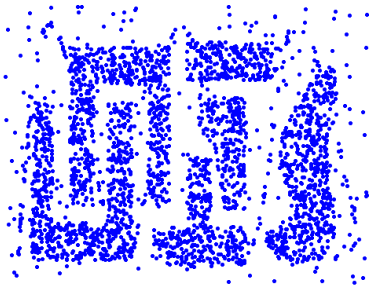
DBSCAN

- DBSCAN chequea la densidad de cada punto en la base de datos contando el número de puntos que contiene cada uno de ellos dentro de su vecindad
- Se crea un cluster por cada punto detectado como “core object”
- Iterativamente colecciona puntos directamente alcanzables por densidad desde los “core objects” agregándolos al cluster, esto lleva a mezclar algunos clusters que contenían TODOS sus puntos directamente alcanzables por el otro
- El proceso termina cuando ningún punto puede ser agregado a ningún cluster

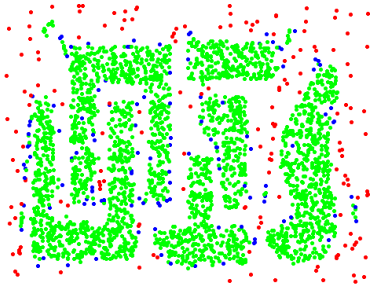


Ejm de DBSCAN

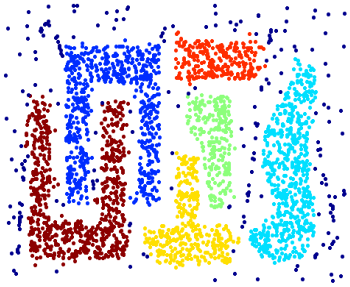
Se ve la arbitrariedad de la forma de los clusters



Datos originales

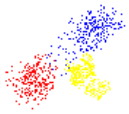


Por tipo: **core**, **border** and **noise**



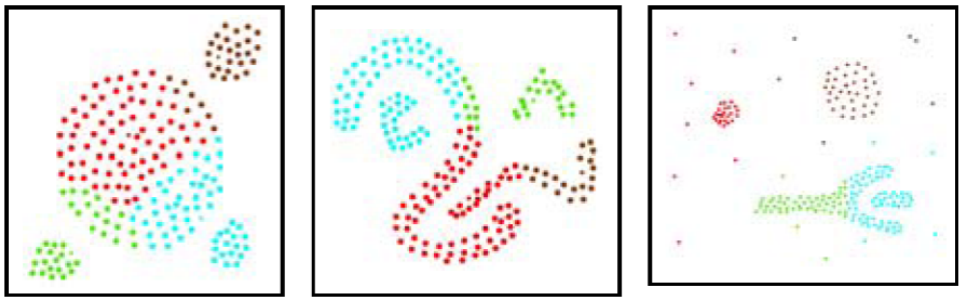
Clusters

Eps = 10, MinPts = 4

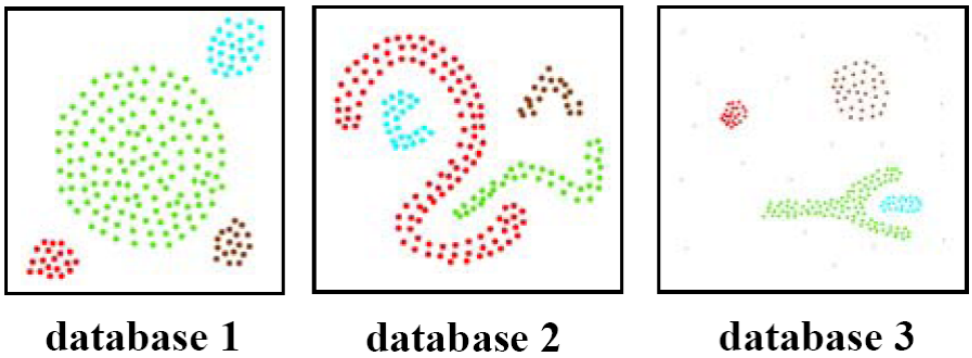


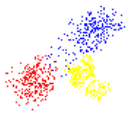
DBSCAN

algoritmo de clustering tradicional



DBSCAN





Problemas de DBSCAN

- Cuando datos tienen regiones de distinta densidad.
- Cuando son datos en altas dimensiones.
- Determinar eps y minpts.

Una forma es usar intuicion que los puntos dentro de un cluster deberian tener la misma distancia al kvecino mas cercano.